INTERFACE AGE™

MICROCOMPUTING FOR SMALL BUSINESS AND HOME VOLUME 3, ISSUE 2 FEBRUARY 1978 $2.00
CANADA/MEXICO $2.50 INTERNATIONAL $3.50

LSI-11 Microcomputers in Hospital ICUs

A Keyboard/Display Interface Peripheral on a Chip

TAPEMON: 8080 Binary Tape Monitor

SFS WALLETSIZE
Spaceship Simulator — Part I

VERSATILITY - UNMATCHED

by any other micro-computer system

COMPUTERS	TAPE SYSTEMS	MEMORY – 4K, 8K, 16K & 32K
TERMINALS	DISK SYSTEMS	PROGRAMMABLE REAL TIME CLOCK
PRINTERS	GRAPHICS	CALCULATOR INTERFACE
SOFTWARE	PROM PROGRAMMER	A - D CONVERTER

Write, or circle our reader reply number for a catalog describing our complete line of computer hardware and software.

NOW AVAILABLE IN BOTH KIT AND ASSEMBLED FORM.

SWTPC

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216

CIRCLE INQUIREY NO. 50

VOL. 3, ISSUE 2
SINCE DECEMBER 1975

FEBRUARY 1978

INTERFACE AGE™

MICROCOMPUTING FOR SMALL BUSINESS AND HOME

This month's cover, and cover story "SFS Walletsize Space Simulator," depict the wonder of space flight and entering the unknown. Space is the final frontier, and is being explored by several nations. Now through the magic of the "micro," and your own imagination, the stars are only a 'byte' away.

FEATURES

HARDWARE FEATURES

SOFTWARE FEATURES

INTERFACE AGE Magazine, published monthly by McPheters, Wolfe & Jones, 16704 Marquardt Ave., Cerritos, CA 90701. Subscription rates: U.S. $14.00, Canada/Mexico $16.00, all other countries $24.00. Make checks payable in U.S. funds drawn on a U.S. bank. Opinions expressed in by-lined articles do not necessarily reflect the opinion of this magazine or the publisher. Mention of products by trade name in editorial material or advertisements contained herein in no way constitutes endorsement of the product or products by this magazine or the publisher.

INTERFACE AGE Magazine is catalogued in the Library of Congress, Classification No. QA75.5.155. Membership in Audit Bureau of Circulations applied for.
POSTMASTER: Please send change of address form 3579 and undelivered copies to **INTERFACE AGE** Magazine, 16704 Marquardt Ave., Cerritos, CA 90701: Second-class postage paid at Artesia, California 90701 and at additional mailing offices.

EDITOR'S NOTEBOOK

By Carl Warren

Evaluating editorial material submitted to a technical trade publication presents a paradoxical problem. The majority of articles are prepared by authors who have a sound technical background, but very little writing experience. The problem for an editor is to take a good technical article and format it into an easily understood manuscript. However, a problem arises when the primary material is presented in such a manner as to make it difficult to edit.

Preparing a manuscript for publication is not as difficult as it probably appears. However, it is necessary to use some very definite steps in the story preparation. Merl Miller, in his Programming the Human Computer column, covered the essentials of article writing in the October and November issues. Therefore, I recommend that you reread these issues. Obviously my main concern is how the manuscript is presented to INTERFACE AGE, or for that matter any publication.

An article accepted by INTERFACE AGE is paid at the rate of 15 to 50 dollars a typeset page. The copy, photos, listing, and line drawing *are all counted* when calculating the article value. The page price is determined by the interest level and counting the number of errors found in the total manuscript. However, before any of this takes place the article must be read and evaluated, and this only happens when the manuscript is presented in a formalized fashion.

The method that we would prefer any prospective author to use is as follows: We need a cover letter telling us about you and your area(s) of expertise. Next an abstract of your article, a short one paragraph summary of what you are going to say. Then it is important to include the outline you used when preparing the story. The next item should be the body of the article with your name, address, and telephone number in the upper left hand corner of each page. The article must be typed, double-spaced on 8½ x 11 white paper. We will not accept articles handwritten or computer printouts, all upper case upsets our typesetters. Then a list of any figures, photos, tables, or listing giving the number and any descriptions or credits that are important. These figures, photos, tables, or listing must then appear on separate sheets and not within the body of the article. Using this method of manuscript preparation will insure that your article will be evaluated, and adds five dollars to the per page rate.

When preparing a hardware article, contact the manufacturer for photos and schematics; most companies are willing to provide them. For articles on something you designed, draw the schematic on graph paper in black ink.

Software articles are unique and present a different sort of problem. Therefore, in order to evaluate them correctly a logic flow chart, drawn on graph paper or prepared with Fickled Thinking Aids, a complete listing, a sample run, and finally a tape or disc of the program, which we will replace, is required.

Unfortunately I am unable to cover everything in detail, but to solve this problem we will send you a style guide kit which covers everything needed to prepare a manuscript correctly. To order the kit write to: INTERFACE AGE, Attn Carl Warren, 16704 Marquardt Avenue, Cerritos, CA 90701.

INTERFACE AGE™

16704 Marquardt Avenue, Cerritos, CA 90701
(213) 926-9544

PUBLISHER & EDITOR-IN-CHIEF **ROBERT S. JONES**

EXECUTIVE PUBLISHER **NANCY A. JONES**

GENERAL MANAGER **EVA YAKA**

MARKETING MANAGER **MEL MURCHISON**

PUBLICATION DIRECTOR **MIKE ANTICH**

SENIOR EDITOR **CARL DENVER WARREN**

NORTHWESTERN REGIONAL EDITOR **ADAM OSBORNE, PhD**

NORTHEASTERN REGIONAL EDITOR **ROGER C. GARRETT**

SOUTHEASTERN REGIONAL EDITOR **BILL TURNER, WB4ALM**

SOFTWARE EDITOR **ABRAHAM A. PEREZ**

HARDWARE EDITOR **ROGER EDELSON**

CONTRIBUTING BUSINESS EDITOR **BUD SHAMBURGER**

EDITORIAL ASSISTANT **DENISE JACKSON**

ASSISTANT-TO-THE-PUBLISHER **KATHY ROBERTSON**

CIRCULATION **DONNA YOSHIOKA**

ACCOUNTING ASSISTANT **SAYOKO TANISAKI**

ART **MERRILYN JOYCE, MARGARET FENSTERMAKER**

PRODUCTION ARTIST **SHELLEY WRIGHT**

SUBSCRIPTION CIRCULATION **JO ANN FERGUSON**

TYPOGRAPHER **MELODY A. MARTENS**

Editorial Correspondence
Direct all correspondence to the appropriate editor at: INTERFACE AGE Magazine, P.O. Box 1234, Cerritos, CA 90701.

NATIONAL SALES MANAGER **BRUCE BERKEY**

ADVERTISING-Representatives **ALLEN BROWNE**
ZACH BOVINETTE

Advertising Inquiries
Direct all advertising inquiries to: Advertising Dept., INTERFACE AGE Magazine, 61 South Lake Avenue, P.O. Box 4566, Pasadena, CA 91106. (213) 795-7002.

DOMESTIC RETAIL CIRCULATION **ZACH BOVINETTE**
(213) 795-7002

JAPAN CIRCULATION **KAZUHIKO NISHI**
ASCII Publishing - 305 HI TORIO, 5-6-4 Minami Aoyama, Minato-ku, Tokyo 107 Japan
Telephone: (03) 407-4910

UNITED KINGDOM CIRCULATION **VINCENT COEN**
L.P. Enterprise, 313 Kingston Road, Ilford, Essex, England IGIIPJ.
Telephone: 01-553-1001

WESTERN CANADA CIRCULATION **BRIAN I.J. WIEBE**
Kitronic, 25236 26th Avenue RR5, Aldergrove, B.C. VOX 1A0
Telephone: (604) 856-2301

EASTERN CANADA CIRCULATION **LIZ JANEK**
RS-232, Ltd., 186 Queen Street West, Toronto, Ontario M5V 1Z1
Telephone: (426) 598-0288

AUSTRALIA CIRCULATION **R. J. HOESS**
Electronic Concepts Pty. Ltd., 52-58 Clarence Street, Sydney NSW 2000
Telephone: 29-3753

MEMBER OF THE WESTERN PUBLISHERS ASSOCIATION

AMERICAN SOCIETY OF BUSINESS PRESS EDITORS

CIRCLE INQUIRY NO. 6

LETTERS TO THE EDITOR

Attention: Robert S. Jones

Subject: More Biorhythm Corrections

Dear Mr. Jones:

This letter supersedes my previous letter, which superseded a previous letter, which . . .

It's been said that any program exceeding ten lines of source code can be counted on to contain at least one bug which will surface sooner or later — probably later. As discussed in my previous letters, the published version of my Biorhythm program contained several minor bugs, some of which were in the source listing I sent you and some of which were apparently introduced somewhere in the publication process. My earlier letters provided fixes for these bugs, requesting that you publish the fixes.

Yesterday I received a phone call from Mr. Jim West, an alert reader, who pointed out that the program does not properly evaluate or act on Leap Years. Looking over the program, I find that Mr. West is absolutely correct. I have modified the program to fix this problem and enclose a listing of the modified program. Some sections of this corrected program are now substantially different from the published version, probably too different to treat effectively by just publishing a change notice. I'll be happy to send corrected listings to any interested readers if they will send me a stamped and addressed return envelope.

William T. Mitchell
5805 Farnham Court
Hanover Park, IL 60103

Dear Editor:

In reference to your Biorhythm program by W.T. Mitchell, page 138, Volume 2, Issue 11, October 1977. . . I found this program like others in that it was written for the money, not for the value of its use. I admit that the program could be of use if it worked, but it does not and could not from the article as written. Either it was taken from a program that did work, but was not copied completely or not understood. Let's see what has happened.

To start off with P1 and P2 were not defined at the beginning, nor was 0$(X). Sub 700 is, as written, rubbish; as was Sub 900. Here we take X and give it a value, go through a bunch of garbage, compare it with itself and if it is still the same send it back to use. Line 605 says GOTO 680 . . . no such line. This would indicate right there that either the program was incomplete or was not printed in its entirety.

Do you check these programs for usability or just like to fill space? If you want to fill space, I can send you several programs that did not work for me. I don't claim to be an expert, but I think that you will see my point. Find this 540 0$(X) = 'P'. Then again 555 and 570 and 580 . . . 585 prints 0$ but what happened to 0$(X)?

Put this program in your computer and see what you get. I have learned from honest programs and have found many good things in your magazine, but this was a low blow. I think that unless readers bring these things to your attention that your quality and reliability as a source of information would begin to be questioned.

Now that that's said, here is something that I found to be of interest and you might want to pass it on to your readers.

ASCII CONTROL CODES

```
CHR$(1 ) = CONTROL A
CHR$(2 ) = CONTROL B
CHR$(3 ) = CONTROL C
CHR$(4 ) = CONTROL D
CHR$(5 ) = CONTROL E
CHR$(6 ) = CONTROL F
CHR$(7 ) = CONTROL G
CHR$(8 ) = CONTROL H
CHR$(9 ) = CONTROL I
CHR$(10 ) = CONTROL J
CHR$(11 ) = CONTROL K
CHR$(12 ) = CONTROL L
CHR$(13 ) = CONTROL M
CHR$(14 ) = CONTROL N
CHR$(15 ) = CONTROL O
CHR$(16 ) = CONTROL P
CHR$(17 ) = CONTROL Q
CHR$(18 ) = CONTROL R
CHR$(19 ) = CONTROL S
CHR$(20 ) = CONTROL T
CHR$(21 ) = CONTROL U
CHR$(22 ) = CONTROL V
CHR$(23 ) = CONTROL W
CHR$(24 ) = CONTROL X
CHR$(25 ) = CONTROL Y
CHR$(26 ) = CONTROL Z
CHR$(27 ) = CONTROL (
CHR$(28 ) = CONTROL
CHR$(29 ) = CONTROL )
CHR$(30 ) = CONTROL †
CHR$(31 ) = CONTROL †
CHR$(32 ) = CONTROL SPACE
```

```
0253 DIM Z$(32)
0254 PRINT
0255 DATA A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,
          Q,R,S,T,U,V,W,X,Y,Z
0256 DATA (,"  ",),†,†,SPACE
0257 FOR X = 1 TO 32
0258 READ Z$(X)
0259 NEXT X
0260 PRINT TAB(15);"ASCII CONTROL CODES"
0262 PRINT
0264 FOR X = 1 TO 16
0266 PRINT "CHR$(";X;") = CONTROL ";
     Z$(X);TAB(28);
0268 PRINT "CHR$(";X + 16;") = CONTROL ";
     Z$(X + 16)
0270 NEXT X
```

Hope someone can use this as it helps to keep this list handy when writing programs.

Robert H. Schwartz
Jeffersonville, IN

Bob, you brought up some valid points that deserve to be addressed. Yes, there were errors in the program that we failed to catch, and several readers have advised us. Also, we do make every effort to insure the reliability of the software before publication. However, in this case we fell down on the job and at this point can only apologize.

We appreciate it when the readership does take the time to point out errors to us. After all without your input we are unable to make the needed corrections. —Editor

Dear Editor:

I think your publication is one of the finest I subscribe to. Your articles and innovative concepts all serve to assist the hobbyist in his or her avocation.

I have only one suggestion: When reprinting programs either enlarge the print from the original or provide a tape copy (paper or mag). It is impossible, unless a magnifying glass is used, to read the code for the HFS I and HFS II programs.

Both of these are exceptional applications and I would like a copy. I

do not, however, desire to become blind from trying to read the reprints.

Garry B. Viele, Jr.
Flint, MI

We have forwarded Mr. Viele's letter to the author for personal reply. —*Editor*

Dear Editor:

I have an Alpha Micro 56K system that I want to get good business programs for. Good systems which are available for very moderate prices I am interested in.

I realize that I will have to convert these systems to Alpha Disc BASIC and would share these converted programs with the original authors. Please help me find the appropriate contacts. Thanks millions.

Scott Brim
7952 Secretariat
Las Vegas, NV 89119

We are publishing this reader's address in its entirety to facilitate his quest for information. —*Editor*

Dear Editor:

As a group of individuals dedicated to peaceful, personal use of computers, we would appreciate your assistance in promoting, through your fine publication, our existence and desire to be of service to other interested individuals. I am sure you share our interest, as many of us in the Charlotte Computer Society subscribe to your publication.

Rex Eagle
Publicity & Liaison Chairman
Charlotte Computer Society
7225 Brynhurst Drive
Charlotte, NC 28210

Always happy to be of service. Good luck, Chaps! —*Editor*

Dear Editor:

This letter is written in appreciation of your magazine. It is to describe to your readers a product which you have advertised in the past. This is the Sabtronics Model 2000 digital multimeter. I have just completed construction of this unit and after its calibration find it exceptionally well-made and useful. I ordered it by mail with payment on June, 1976. In August, 1976 I received a letter that production would be delayed another 5 to 6 weeks and that I could apply for refund if I preferred. In October, 1976 I called the firm to learn what was happening. The only listed telephone lines were perpetually busy. In late October, I received the second letter that the unit would be shipped in 2 weeks and that I still could apply for refund instead. It arrived in late November.

Construction was straightfor-

ward. The components appeared to be of high quality. The only problems occurring in construction were problems reading the color on one resistor and the back to back placement of two IC units where the one on the right faced right and the one on the left faced left. Compulsion to orient these in the same direction gave unusual readings until the problem was discovered. No harm occurred to the IC. Sockets were not provided for the non critical chips and I opted to install these.

The major problem for me was when I got to the page on calibration. This required some voltage and current sources which were not immediately available. Fortunately a client operated a calibration lab for a major corporation and was able to do this for me. The alternate source available would be another well-calibrated digital multimeter. With this the sources could be easily arranged and the trim pots properly set.

I read your magazine and particularly its advertisements and new product descriptions avidly. I am attempting to develop a rather complex computer system for a special business application. I would like to read more about letter quality printers.

Keep up the good work.

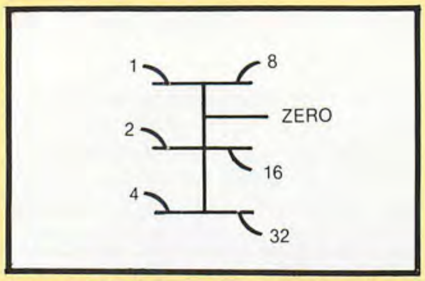
Robert C. Luckey, M.D.
1110 Gillmore Ave.
Richland, WA 99352

We are publishing Dr. Luckey's address in the hope that readers and manufacturers will respond to his quest for information. —Editor

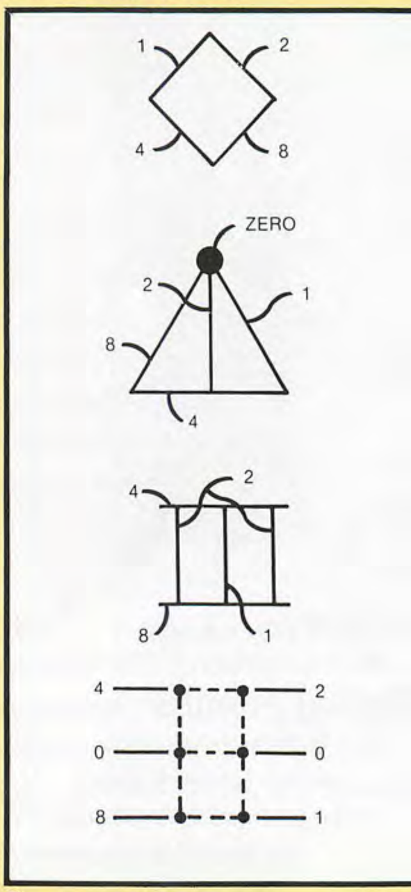
Dear Editor:

Irwin Doliner in an article in your November 1977 issue discusses the several numbering systems of interest to computerists. It is noted that he follows IBM in using letters of the alphabet to fill the positions 10 to 15 in his hexadecimal set. This leads to confusion. Does BAD mean bad? Or does it represent a number? If the latter, it is still bad. Or worse.

From time to time people have proposed "Computer Compatible Digits" which avoid the confusion brought on by IBM's desire to avoid introducing new numerals which would require additional hammers on their printers. A gentleman in Prague in the fifties proposed:



A base-64 digit. Each numeral of the set may be formed of a properly chosen subset of the weighted lines. No encoder going to or from a computer. Since then the following have been proposed:



The cost saving and the elimination of confusion appear to be sufficient to justify our change to some such "Computer Compatible Digit". There are only a few things more asinine than using letters of the alphabet as numerals.

R. O. Whitaker
Indianapolis, IN 46241

Dear Editor:

In your library of computer programs, do you happen to have a program with which one could simulate

CIRCLE INQUIRY NO. 2

the game of Contract Bridge on a computer, preferably an economical micro computer type. If not, do you know of any company that might have such a program available? Thank you.

William F. Troupe
8477 Deauville
Pinellas Park, FL 33565

We are publishing this reader's address. We don't have one here, but by chance someone might oblige him. We would also be interested in publishing such a program.
—Editor

RE: THE ENERGY/ENVIRONMENT SIMULATORS, OCTOBER 1977

Dear Editor:

We would like to clear up some confusion which resulted from editing in the article, "The Energy/Environment Simulators," published in your October 1977 issue. In the third paragraph of the article, explanations about the regional simulator appear to be describing the national simulator.

The national simulator consists of a single console with several lap panels which allow users to change variables during simulations. All computing capability and displays are contained in the main console. The regional simulator consists of seven consoles. Six of these are diagrammed in Figures 1a-1f on page 58. The seventh is a larger, main display console. Computing capability, displays and variable input are distributed among these consoles.

The national simulator was developed by Dr. John Amend under contract to the Atomic Energy Commission. The regional simulator was developed jointly by Dr. Amend and the Montana Energy and MHD Research and Development Institute (MERDI).

At the end of the article it is mentioned that additional information can be obtained from the Office of Public Affairs, ERDA (now DOE). They have information on the education program they sponsor, which uses the national simulator. Requests for additional information about the regional simulator should be addressed to Tom Pelletier, MERDI, P.O. Box 3809, Butte, MT 59701.

Byron Jones, Sr. Systems Analyst
MERDI
Joan Melcher, Writing Consultant
MERDI

Before publication I read the rewritten text to Ms. Melcher and mailed a check copy of the manuscript. Dr. Amend's corrections were included in the published text. *—Editor*

Dear Editor:

For your next Floppy ROM not spoken for, why not a nice 8080 chess program? It should be written in a BASIC everyone can use, and should not take much more free space than most people have for program and execution, say 8K. I'm

The Dumb Terminal lets you put it all together.

With the new, lower-priced Dumb Terminal™ Kit, that is. Pick one up and escape, once and for all, the headaches of scavenged teletypes and jury-rigged TV sets. With just a little time and aptitude, you can have a live and working Dumb Terminal right in your own home, garage, or business. One that lets you get it all out of your system — or into it.

Forget the cheap imitations, with their overblown price tags and interminable lists of options. With the Kit, you can build yourself the same, old basic Dumb Terminal that's been selling over 1500 units a month. With basic, sensible features like a bright 12" diagonal screen. Fifty-nine data entry keys. 1920 characters displayed in 24 rows of 80 letters. Plus 33 positive action switches that let you activate functions like 1 of 11 different baud rates, an RS232C interface, or a 20mA current-loop. And more. Not bad for Dumb.

All you need, besides the Kit, is some initiative, and a few basic tools — a good soldering iron, wire cutters, needle-nose pliers, and one or two trusty screw-drivers. The Dumb Terminal Kit provides you with every-thing else. Including an attractive cabinet, CRT screen, keyboard, PC board, and all essential electronic components. Naturally, you also get illustrated, step-by-step assembly instructions, not to mention an easy-to-understand operator's manual.

So, if you'd like more input on the Dumb Terminal Kit, just fill out the coupon and we'll send you complete, free information.

Oh, and by the way, just by sending in the coupon, you will be made a charter member of the Dumb Terminal Fan Club. A select organization that will send you your own nifty Dumb Terminal Fan Club Kit, containing: an official certificate of membership; an autographed photo of the Dumb Terminal himself; and a bona fide membership card to prove irrefutably you're "One of Us." (Sorry, limit one kit per person.)

And, if you include a trifling $6.00, you can have your very own Dumb Terminal T-shirt. (No limit at all on these.)

Simply mail the coupon and get the whole assortment. And find out why members of the Dumb Terminal Fan Club are some of the smartest people around.

LSi

Dumb Terminal. Fun Club.

Yes, I would like more information about the incredible Dumb Terminal Kit. And don't forget to enroll me as a Charter Member in the Dumb Terminal Fan Club.

Name________________ Title(?)________

Company________________

Street________________

City__________ State_____ Zip_____

Enclosed is $________ for an official Dumb Terminal T-shirt(s).

Please make all checks and money orders payable to LeAnce & Reiser, OK?

Quantity and size(s) of shirt(s) required: ____S ____M ____L ____XL

Rush this application to: Dumb Terminal Fan Club Headquarters c/o LeAnce & Reiser, P.O. Box 17123, Irvine, CA 92714.

"Dumb Terminal" is a trademark of Lear Siegler, Inc., E.I.D./Data Products, 714 N. Brookhurst St., Anaheim, CA 92803, (800) 854-3805. In California (714) 774-1010.

CIRCLE INQUIRY NO. 26

sure it would be the most popular FR to date.

Thank you.

Gary Martin
Gladstone, MO

Gary, you have an excellent idea. Watch for our August issue which will be filled with games of all types, and a Floppy ROM of the more exciting ones such as chess. —Editor

Dear Editor:

I hope you remember me favorably regarding the article on speech recognition in May last year. The printing was excellent, exactly as I had written. The reader responses were interesting and complimentary.

Owen F. Thomas
La Jolla, CA

It is nice to know that our authors like our work. Thanks, Owen. —Editor

 CIRCLE INQUIRY NO. 31

FLOOR PLAN

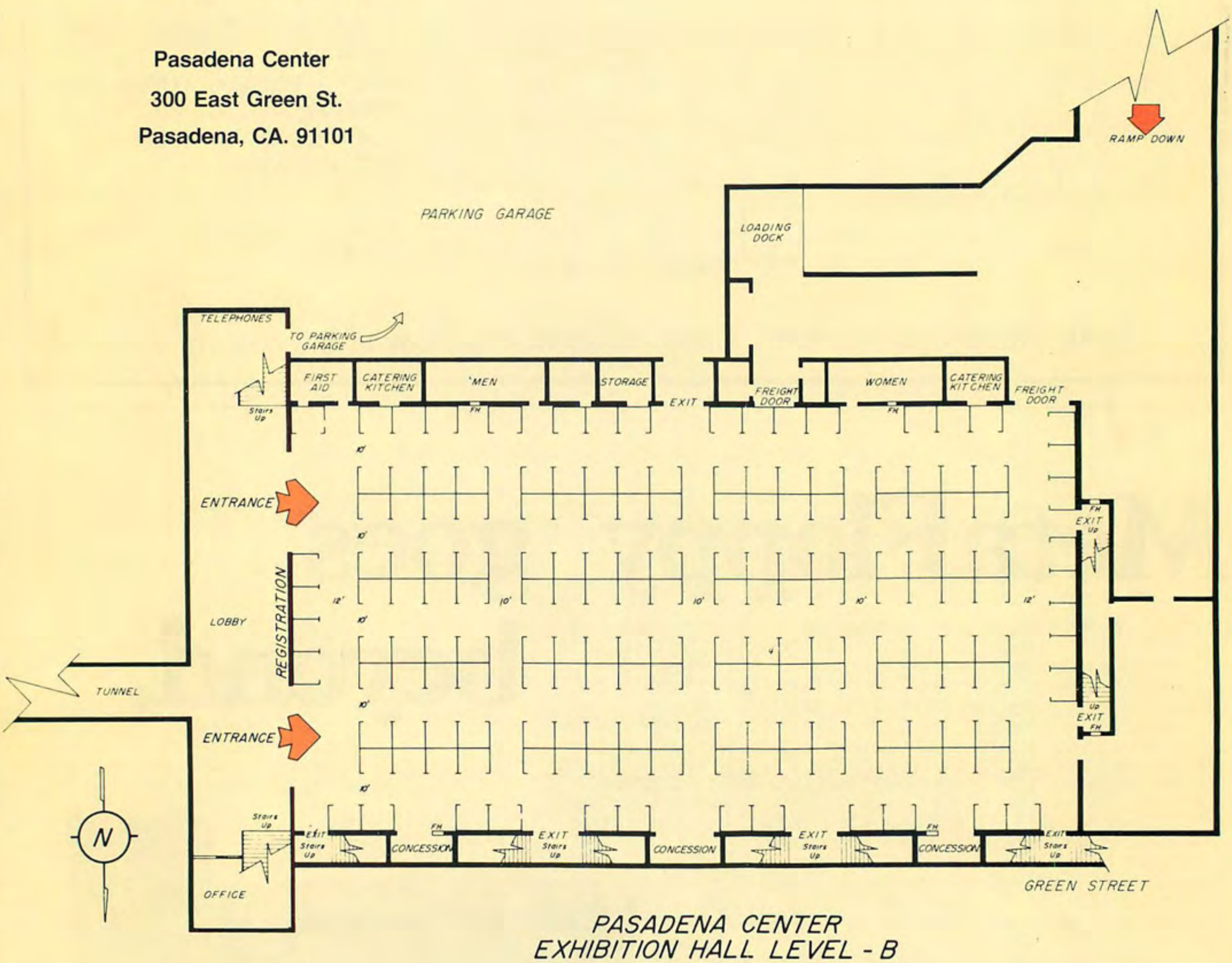
MICRO BUSINESS '78

March 17, 18, 19, 1978

Pasadena Center

300 East Green St.

Pasadena, CA. 91101



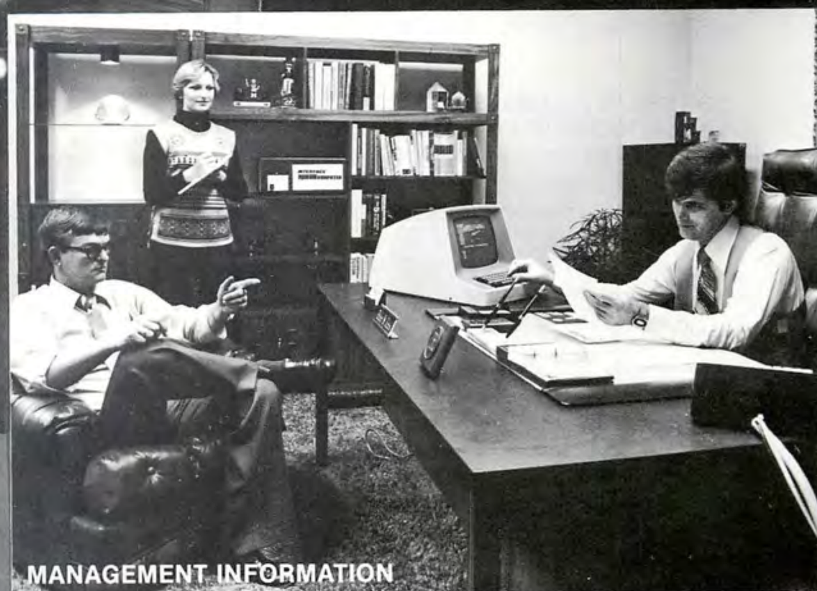
PASADENA CENTER
EXHIBITION HALL LEVEL - B

MICRO BUSINESS '78™

DATE: MARCH 17, 18 & 19
PLACE: PASADENA CONFERENCE CENTER

COMPLETE DISPLAY OF LOW COST BUSINESS COMPUTER SYSTEMS

CONSUMER SHOW



MICRO BUSINESS '78 Consumer Show will be a complete exhibition of the new affordable microcomputer systems, directed to the small business environment.

Emphasis will be on the low cost, dependability, simplicity of operation and cost saving advantages of a microprocessing system.

The exhibits and conferences at MICRO BUSINESS '78 will provide the information required to help you understand the microcomputer field and how it relates to your business.

Take the time to learn why more and more people are getting involved in the microcomputer business. Allow MICRO BUSINESS '78 to show you, the small businessman, that a microcomputer is an investment. . . not just an expense.

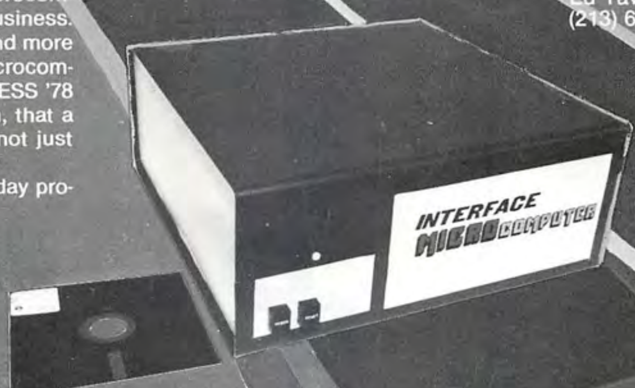
PRE-REGISTER now for the three day program and lecture series.

TICKET INFORMATION

Pre-registration tickets can be purchased for $12.00 for the three day exhibition. To order tickets, send check or money order to Show Company International, 8687 Melrose Avenue, Los Angeles, California 90069. Pre-registration tickets will be held in will call at the show.

Tickets can also be purchased on the show dates for $10.00 per day.

For further exhibit information contact:
Ed Tavetian, Show Company International,
(213) 659-2050.



PRODUCED BY SHOW COMPANY INTERNATIONAL
SPONSORED BY INTERFACE AGE MAGAZINE

UPDATE

PERSONAL COMPUTING FESTIVAL

The National Computer Conference '78 presents the Personal Computing Festival scheduled to take place at the Anaheim Convention Center, June 5-8, 1978.

Featured will be a large selection of computing products and a number of interesting conference themes such as graphics, music systems, speech synthesis, small business systems, microprogramming, computer games, personal bionic systems, software design, optical scanners, and standards.

For registration and further information write to NCC '78 Personal Computing Festival, 210 Summit Ave., Montvale, NJ 07645, or call Marie Stewart at (201) 391-9810.

COURSE IN DIGITAL ELECTRONICS & MICROCOMPUTER INTERFACING

Virginia Military Institute will conduct a two-week course in digital electronics and microcomputer interfacing from July 17 thru July 29, 1978.

This is a hands-on course designed for both academic and industrial personnel who are interested in the implementation of microcomputer techniques to solve problems in computer interfacing. Lecture and laboratory topics will include instruction in the fundamentals of digital electronics, the 8080 microprocessor and standard interfacing techniques. Software development aids will be available.

This course provides an opportunity for professional growth in this popular technological area as well as a vacation in the historic and beautiful Shenandoah Valley of Virginia. Tuition will be $350.00, of which $100.00 deposit will be required by April 5, 1978. Academic credit is available through James Madison University.

For information and registration forms write to Dr. Philip B. Peters, Dept. of Physics, VMI, Lexington, VA 24450.

CALL FOR PAPERS

1978 International Optical Computing Conference, September 5-7, London, England. Papers on advanced optical signal processing systems, applications, and technology are solicited for this conference sponsored by the IEEE Computer Society in cooperation with SPIE and the U.S. Office of Naval Research.

Principal topics are: radar and sonar signal processing, near field and far field pertubations, electronic to optical signal conversion, acoustical to optical signal conversion, hybrid optical/electronic/digital systems, applications of non-linear and space-variant optical systems, and advanced thin film and integrated optics signal processing systems.

An abstract of 200-250 words plus a brief biographical sketch should be submitted by April 1, 1978 to either E.C. Landau, Naval Underwater Systems Center, New London, CT 06320 or LCDR David Rummier, Office of Naval Research, London, 223 Old Marylebone Rd., London, NW1, Great Britain.

NEW PROFESSIONAL ASSOCIATION

The Southern California Computer Dealers Association has been formed, as a non-profit corporation, covering the Santa Barbara to San Diego area. Membership is limited

to bona-fide stores which sell microcomputers for personal, business, industrial or educational use.

The organization will have dinner meetings on the third Monday of each month. Visiting dealers and microcomputer industry leaders are cordially invited to the dinner meetings.

The meetings will offer a forum for the exchange of views and information about the industry. The association is setting up committees to investigate further activities and objectives such as publicity for microcomputers, professional standards, and dealer-vender relationships.

For further information, contact Carl Burlin at (714) 956-7968, or Glenn Dollar at (213) 360-2171.

1978 BUSINESS AND PERSONAL COMPUTER SALE & EXPOSITION

Under the aegis of Produx 2000, Inc. an exhibition of new equipment and specialized products of interest to computer users will take place at the City Line Holiday Inn, Philadelphia, PA, from February 27 to March 1, 1978.

For further information, write to Produx 2000, Inc. Convention Services Dept., Roosevelt Blvd. and Mascher St., Philadelphia, PA 19120, or call (215) 722-4837.

COMPCON SPRING '78

"Computer Technology: Status, Limits, Alternatives" is the theme for COMPCON '78 Spring, to be held February 28-March 2, 1978, at the Jack Tar Hotel, San Francisco, California. A pre-conference tutorial on "Limitations and Alternatives in Future Silicon LSI Technology" will be offered, as well as daytime technical sessions covering three major areas, and a short notes session.

For the first time in the 16-year history of COMPCON, personal computing sessions, exhibits, and demonstrations will be featured. Subjects to be covered during these evening sessions include: "Women's Role in Innovative Computer Applications," "Robotics and Bionics" (including computer aid for the severely handicapped), and "Computers in Art and Music." For further information contact COMPCON '78 Spring, P.O. Box 639, Silver Spring, MD 20901, (301) 439-7007.

INTERFACE '78

Data Communications in co-sponsorship with Datamation Magazine will present the Sixth Annual Conference and Exposition at the Las Vegas, Nevada Convention Center, March 6-9, 1978.

Featured will be conferences and speakers. Booth space is still available. For reservations or further information, write to Data Communications Interface '78, 160 Speen St., Framingham, MA 01701, or call toll-free (800) 225-4620.

ELECTRO '78

An electronic show and convention will take place at Hynes Veterans Auditorium, Boston, MA, May 23-25, 1978.

Represented will be exhibitors from the following industries: machine tools, office and computing machinery, instrumentation and measurement, medical and surgical equipment, and communications.

CIRCLE INQUIRY NO. 18

The program will be held at the Sheraton Boston Hotel.

For reservations and further information on booth rental or attendance, write Electro '78, Suite 410, 999 N. Sepulveda Blvd., El Segundo, CA 90245 or call toll-free (800) 421-6816 (in California call collect (213) 772-2965) or TWX 910 348-7127.

PERCOMP '78

Eight free seminars, tutorials and demonstrations are scheduled for PERCOMP '78, April 28-30 at the Long Beach Convention Center, Long Beach, California.

The papers run the gamut from the very basic to the super sophisticated, and also encompass the lighter aspects of computer games, music and household use.

Included are the following presentations: "Marketing for the New Manufacturer," David Ahl, Creative Computing; "Three Dimensional Microcomputer Graphics," Bruce Artwick, Sublogic; "6530 Timer Programming," Arthur Stoll, Rockwell International; "Human Factors in Software Design," Jack Emmerichs, A.O. Smith; "Computer Games," James Butterfield, author of *The First Book of KIM*; "Getting Started in Microcomputers," Louis Fields, president, International Computer Society/SCCS; "The 'Jogger' Microprocessor Communication Bus," Dr. Keith L. Dotty, University of Florida.

Dr. Portia Isaacson, chairperson for the ACM Personal Computing Group, will conduct a session for retailers, while attorneys Leonard Tachner and Kenneth Widelitz will take computer enthusiasts through the labyrinth of tax benefits, patents and copyrights.

Carol Ogdin, Software Technique, Inc., will deliver the keynote address. She will address the subject of proven home applications of the computer.

For further information, write PERCOMP '78, 1833 E. 17 St., Santa Ana, CA 92701.

ISRATECH '78

American executives visiting IsraTech '78 will meet in Israel with the Minister of Industry, Commerce and Tourists and with the Mayor of Jerusalem as part of the schedule of events announced by the Government of Israel Economic Offices.

IsraTech '78, an exposition of Israel's rapidly growing high technology industries, is scheduled for June 4th to 8th, 1978 in Jerusalem. It will be the most significant economic event of Israel's thirtieth anniversary.

In addition to meeting with Minister Hurvitz and Mayor Kollek, attending executives will meet with Israel's manufacturers and government representatives to discuss a vast number of business opportunities in Israel, such as buying, selling, investing and licensing. Executives will also attend an International Meeting of the Metalworking and Professional Electronics Committee, will tour industrial plants and research and development facilities, and will be guests at official banquets and receptions.

While attending a modern exposition, executives and their families will also be able to visit the sites of one of the world's most historic cities. The program for IsraTech '78 will include tours of the Israel Museum, cultural evenings, and a special program designed for accompanying persons.

At the exhibition visiting executives will also have an opportunity to discuss with their Israel counterparts and with representatives of the Government the unique advantages Israel offers potential foreign investors. Among these benefits are:

- Common Market agreement providing duty-free entry to the 270 million person Common Market by manufacturing for export in Israel.

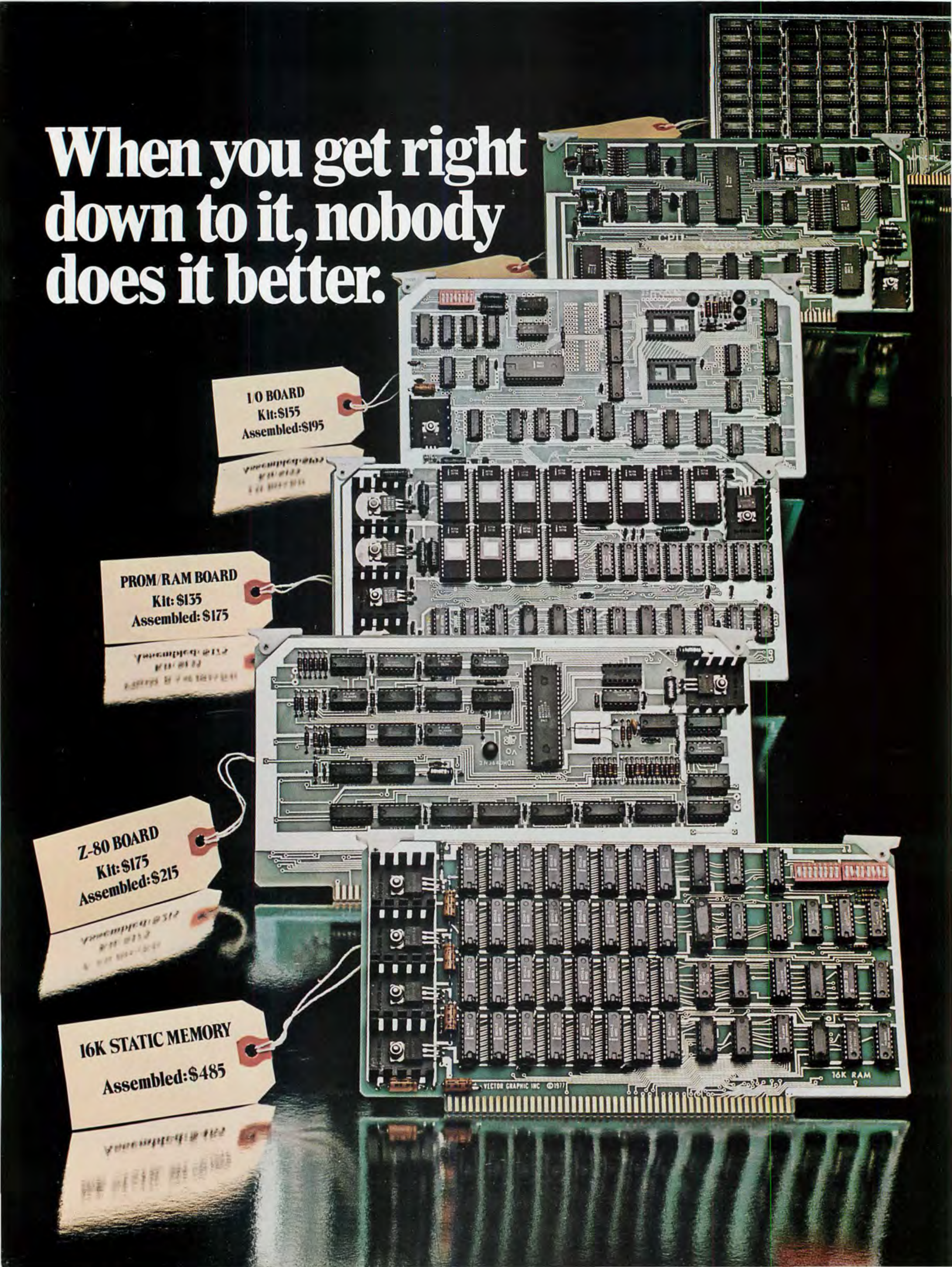
- Israel enjoys the Generalized System of Preferences by which 2,700 Israel-made product categories can be imported to the U.S. duty-free.
- U.S. companies will be exempt from double taxation on profits earned in Israel under a soon to be ratified, and already signed, tax treaty.
- Israel offers a generous package of financial and operating incentives to encourage U.S. companies to locate production facilities there, including financing most of the fixed assets and working capital needed.
- Israel actively supports industrial research and development activities with grants of 50% of R&D costs; moreover, Israel and the U.S. recently established the Binational Industrial Research and Development Foundation, which has been allocated $60 million to fund joint R&D projects.

For further information contact the Government of Israel Investment Authority, 641 Lexington Ave., New York, N.Y. 10022, (212) 486-8530.

"Mr. Bradshaw, there is a Beverage Agent here to see you about your homebrew."


ANNOUNCING!
Software
Development Systems
SPACE BYTE
SPACE BYTE
SPACE BYTE


by SPACE BYTE™
SPACE BYTE Software Development Systems are versatile and cost effective; featuring the SPACE BYTE 8085 CPU, self contained computer, 16K SPACE BYTE fully static RAM and the new SPACE BYTE 2708/2716 EPROM Programmer. All PROM programming, I/O handling FDOS, and complete debugging routines, with breakpoints are intrinsic to the SPACE BYTE system monitor. Complete software support too, with iCOM FDOS III, 8080, 8085, Z-80 relocatable macro assembler, operating system, Disk Extended Basic, CP/M and FORTRAN-80 and the SPACE BYTE exclusive 3870/F8 cross assembler.
THE SPACE BYTE SYSTEMS GROUP
Software Development Systems
Business Systems
Word Processors
SPACE BYTE MICRO COMPUTER PRODUCTS ARE SUPPORTED BY COMPUTER RETAILERS WORLD WIDE.
1720 Pontius Ave. Suite 201
Los Angeles, CA 90025
SPACE BYTE SOFTWARE DEVELOPMENT SYSTEM PRICING STARTS AT $5900.
Special Offer
Save $150.00
Present the special discount coupon (found later in this magazine) to any participating computer retailer, and they will package the system for your particular application. YOU SAVE 10% of the end user price on the combination SPACE BYTE 8085 CPU, 16K SPACE BYTE and 2708/2716 EPROM Programmer (total price $1347.). Offer good at participating computer retailers only, void where prohibited, taxed, or restricted by law. Offer expires March 31, 1978.
CPU • RAM
EPROM
PROGRAMMER


When you get right down to it, nobody does it better.
I/O BOARD
Kit: $155
Assembled: $195
PROM/RAM BOARD
Kit: $135
Assembled: $175
Z-80 BOARD
Kit: $175
Assembled: $215
16K STATIC MEMORY
Assembled: $485
VECTOR GRAPHIC INC ©1977
16K RAM

"12 persuasive reasons why Vector Graphic gives you more microcomputer value for your money."

Bob Harp, Technical Director

1 Our products are designed by professionals with many years of experience in industrial electronics.

2 Only prime components are used from vendors you will recognize as the most reputable in the industry.

3 Our kits are designed to be easily assembled with a minimum of user installed jumpers — you don't have to spend hours figuring how to configure boards for your application.

4 Our cabinets quality is superior — designed with careful attention to airflow (a fan is standard) and mechanical rigidity.

5 We offer a complete line of products designed to operate together — no need to be concerned about compatibility.

6 You have a choice of CPU boards with either 8080 or Z-80 MPU.

7 We offer all of our products assembled and tested — for surprisingly little more than kit prices.

8 Our management is genuinely interested in you as a satisfied customer.

9 You can purchase our products with confidence — they will not become obsolete with the next generation of microprocessors.

10 We offer complete technical support with engineers and technicians available to answer your questions.

11 Our speedy shipment of orders is the envy of our competitors.

12 A network of over 150 dealers throughout the U.S., Canada and Europe has been carefully selected to service your needs.

No other company in the industry has grown as rapidly as Vector Graphic.
If you want to know why Vector Graphic has outstanding acceptance, purchase one of our products.
Write or call for your nearest dealer.

© VECTOR GRAPHIC INC.

Vector Graphic, Inc., 790 Hampshire Rd., A + B
Westlake Village, CA 91361, Tel: (805) 497-6853

A dozen ways you can keep current in microcomputing:

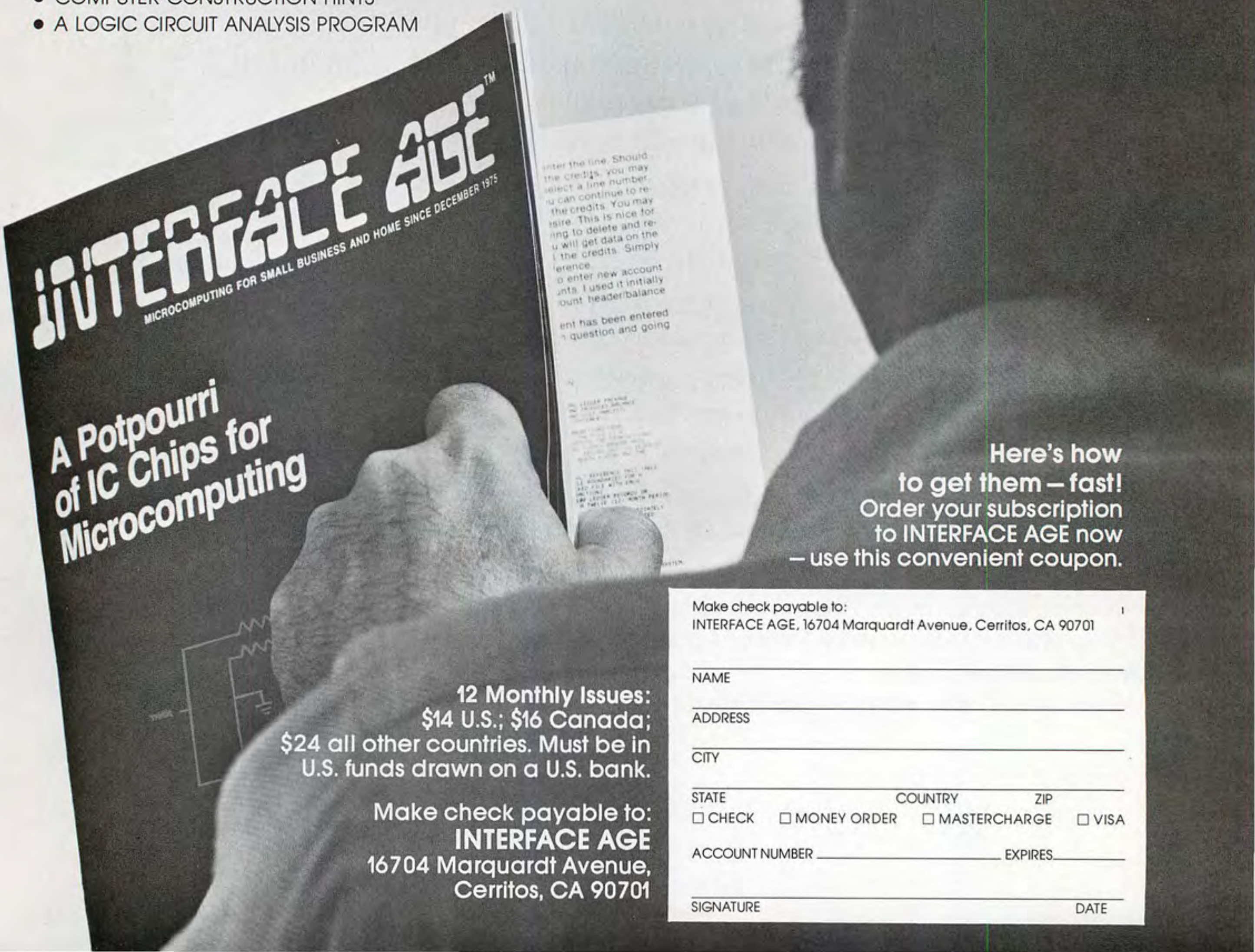
the next 12 issues of Interface Age!

INTERFACE AGE is your kind of publication ... written especially for those involved in microcomputing who want to keep updated on the latest advances in this dynamic new technology.

It's your monthly update on new products, hardware, software, tutorials and articles that speak your language and help you in your job. If you're into computing, INTERFACE AGE is the one magazine you shouldn't miss!

CHECK THESE TYPICAL ARTICLES...

- A POTPOURRI OF IC CHIPS FOR MICROCOMPUTING
- ROBOTICS AND BIONICS—TOMORROW'S TECHNOLOGY TODAY
- HELP YOUR COMPUTER UNDERSTAND YOUR VOICE
- COMPUTER CONSTRUCTION HINTS
- A LOGIC CIRCUIT ANALYSIS PROGRAM



Here's how to get them—fast!
Order your subscription to INTERFACE AGE now —use this convenient coupon.

12 Monthly Issues:
$14 U.S.; $16 Canada; $24 all other countries. Must be in U.S. funds drawn on a U.S. bank.

Make check payable to:
INTERFACE AGE
16704 Marquardt Avenue, Cerritos, CA 90701

Make check payable to:
INTERFACE AGE, 16704 Marquardt Avenue, Cerritos, CA 90701

NAME ______________________

ADDRESS ______________________

CITY ______________________

STATE ________ COUNTRY ________ ZIP ________

☐ CHECK ☐ MONEY ORDER ☐ MASTERCHARGE ☐ VISA

ACCOUNT NUMBER ______________ EXPIRES ________

SIGNATURE ______________________ DATE ________

CALENDAR

MARCH

Mar 1 New England Computer Society will meet in the cafeteria of the MITRE Corp. at 7:00 P.M. Located on Route 62 in Bedford, MA. Contact Dave Day at (603) 434-4239 for details.

Mar 1 Kitchener Waterloo Microcomputer Club will meet at the University of Waterloo, Room 3388, Engineering Bldg. #4, University Ave., Waterloo, Ontario, Canada at 7:30 P.M.

Mar 1 Northwest Computer Society will meet in the Pacific Science Center in Seattle, Room 200 at 7:30 P.M. For more details write NCCN, Box 242, Renton, WA 98055.

Mar 1 The Valley Computer Club will meet at 7 P.M. at the Harvard School located at 3700 Coldwater Canyon, Studio City, CA.

Mar 1 Lincoln Computer Club will hold its meeting at the South Branch Library located on 27th and South Sts. at 7 P.M. For more details write Hubert Paulson, Jr., 422 Dale Dr., Lincoln, NE 68510.

Mar 1 Amateur Computer Society of Columbus will meet the first Wednesday of each month at the Center of Science and Industry at 7:30 P.M. For further information write c/o Fred Hatfield K8VDU, Computer Data Systems, 1372 Grandview Ave., Columbus, OH 43212, or call (614) 488-3347.

Mar 2 Bay Area Microprocessors Users Group (BAMUG) will meet in the Hayward ROC Center, 26316 Hesperian Blvd., Hayward, CA at 7:30 P.M. For further details write BAMUG, 1211 Santa Clara Avenue, Alameda, CA 94501.

Mar 3 Crescent City Computer Club will hold its meeting at the University of New Orleans, Lakefront Campus at 8 P.M. Call Bob Latham at (504) 722-6321 for more details.

Mar 4 Louisville Area Computer Club (LACE) will meet at the University of Louisville, Speed School Auditorium at 1 P.M. For details, write the club at 115 Edgemont Dr., New Alban, IN 47150.

Mar 4 The Computer Hobbyist Group, will meet at 1 P.M. in Green Center, Room 2.530, campus of University of Texas, Dallas. For further information write the club at P.O. Box 11344, Grand Prairie, TX 75051.

Mar 4 South Central Kansas Amateur Computer Association, 9:00 A.M., Wichita Public Library, Wichita, KS. For further information call Chris Borger at (316) 265-1120 or Dave Rawson, 1825 Gary, Wichita, KS 67219, (316) 744-1629 for further details.

Mar 4 Oklahoma Computer Club will be meeting at the Belle Aisle Library at 10 A.M. Call Al Campbell at (405) 842-4933 for details.

Mar 4 Southern Nevada Personal Computing Society will meet at Clark County Community College, Las Vegas, NV at 12:00. For further information write SNPCS, 1405 Lucille St., Las Vegas, NV 89101 or call (702) 642-0212.

Mar 4 Milwaukee Area Computer Club will meet at 1 P.M. at the Waukesha County Technical Institute, New Berlin, WI. Call (414) 246-6634 for further details.

Mar 6 Minnesota Computer Society will meet at the Brown Institute, Room 51, 3123 E. Lake Street, Minneapolis, MN. For further information contact the Society at Box 35317, Minneapolis, MN 55435, Attn: Jean Rice.

Mar 7 Tidewater Computer Club will hold its meeting at the Electronics Computer Programming Institute, Janaf Office Bldg., Janaf Shopping Center in Norfolk. The club also meets on the third Tuesday of the month. For further information contact: C. Dawson Yeomans, Interface Chairman, 677 Lord Dunmore Dr., Virginia Beach, VA 23462.

Mar 9 Mid America Computer Hobbyist meeting will be at 7:00 P.M. at Commercial Federal Savings & Loan, Bellevue NE. Intersection of Galvin Rd. and U.S. Hwy. 73-75. Write P.O. Box 13303, Omaha, NE 68113 for further information.

Mar 9 Utah Computer Association will meet at Murray High School, Rm 154, 5440 S. State St., Salt Lake City, UT at 7 P.M. For details write or call Larry or Holly Barney, 1928 S. 2600 E., Salt Lake City, UT 84108. (801) 485-3476.

Mar 9 The Rochester Area Microcomputer Society will meet at the RIT Campus, Rm. 1030, Bldg. 9 at 7:30 P.M. For details write RAMS, P.O. Box D, Rochester, NY 14609.

Mar 10 Northern New Jersey Amateur Computer Club (NNJACC) will hold its meeting at the Fairleigh Dickenson University, on the Rutherford Campus, Becton Hall, Room B8, at 7 P.M. For details

write NNJACC, 593 New York Ave., Lyndhurst, NJ 07071.

Mar 11 The Permian Basin Computer Group — Odessa Chapter meets at 1 P.M. in the Electronic Technology Bldg., Room 203 on the Odessa College campus. For details call (915) 332-9151.

Mar 12 North Orange County Computer Club will have its meeting at Chapman College, Orange, CA. Doors open at 12:00. 105 Hashinger Hall Auditorium. Membership Chairman, Tracey Lerocker, (714) 998-9722 evenings.

Mar 15 Homebrew Computer Club meeting will begin at 7 P.M. in Menlo Park, CA at the Stanford Linear Accelerator Center Auditorium. Call (415) 967-6754 for more details.

Mar 17 Long Island Computer Association will meet at 7 PM at the New York Institute of Technology, Old Westbury Campus, Route 25A between Route 107 and Glen Cove Rd., Rm. 508. For more details write Long Island Computer Association, 36 Irene Lane East, Plainview, NY 11803.

Mar 17 Amateur Computer Group of New Jersey (ACGNJ) will meet at UCTI, 1776 Raritan Rd., Scotch Plains, NJ 07076 at 7 P.M. For further information write to the club at the above address.

Mar 18 Southern Nevada Personal Computing Society will meet at Clark County Community College, Las Vegas, NV at 12:00. For further information write SNPCS, 1405 Lucille St., Las Vegas, NV 89101 or call (702) 642-0212.

Mar 18 San Diego Computer Society will meet at the Grossmont Community College Student Center, 8800 Grossmont College Dr., El Cajon, CA. Doors open at 12:30. For details call (714) 565-1738.

Mar 18 The 7C's Committee (Affiliated with the Cleveland Digital Group) will meet at Cleveland State University Student Services Bldg., in the Kiva Room at 2:00 P.M. For more information write to Cleveland Digital Group, 8700 Harvard Ave., Cleveland, OH 44105.

Mar 18 Central Florida Computer Club will meet at the Orlando Utility Bldg., on S. Orange Ave., Orlando, FL at 2:00 P.M.

Mar 18 Philadelphia Area Computer Society will meet at 2 PM at LaSalle College Science Bldg. at the corner of 20th & Olney Ave. For more details write PACS, P.O. Box 1954, Philadelphia, PA 19105.

Mar 19 Chicago Area Computer Hobbyist Exchange (CACHE) will meet at 12:00 P.M. in the Nigas Bldg. Cafeteria located on Schermer Rd. in Glenview, IL. Call CACHE Hotline (312) 849-1132 for details.

Mar 21 Sacramento Microcomputer Users Group, (SMUG), 7:30-9:30 P.M. at SMUD Training Bldg., on 59 St. Write Richard Lerseth, P.O. Box 161513 or call (916) 381-0335 after 5:00 P.M.

Mar 21 Rhode Island Computer Hobbyists (RICH) meets the at the Knight Campus of Rhode Island Junior College in the Faculty Cafeteria at 7:30 P.M. For further information contact Emilio Iannucillo, RICH, P.O. Box 559, Bristol, RI 02809, or call (401) 253-5450.

Mar 22 Boston Computer Society will meet at the Commonwealth School, 151 Commonwealth Ave., Boston at 7 P.M. The school is

"What with all the new computers and calculators, I cracked under the pressure of the arithmetic exam and just let down!"

"I couldn't ask for a better location."

located on the corner of Dartmouth St. in Boston's Back Bay. For information write or call the society at 17 Chestnut St., Boston, MA 02108, (617) 227-1399.

Mar 22 Diablo Professional Users Group (DPUG) will meet at Diablo Valley College Library, near the Willow Pass exit of Fwy. 680, from 8-10 PM. For details write or call Bob Hendrickson, Electronics Dept., DVC, Pleasant Hill, CA 94523; (415) 687-8373.

Mar 24 TRACE will hold its meeting at the Ontario Science Center, 8 P.M., 770 Don Mills Road, Don Mills, Ontario. Club address is Box 545, Streetsville, Ontario, Canada L5M 2C1.

Mar 24 Alamo Computer Enthusiast meets at 7:30 P.M. in Room 104 at Chapman Graduate Center at Trinity University, San Antonio, TX. For details call (512) 532-2340, or write to the club at 7517 Jonquill, San Antonio, TX 78233.

Mar 26 Summit City Computer Club will meet at the McMillen Library on the Indiana Institute of Technology Campus in Ft. Wayne, IN. For details write the club at P.O. Box 5096, Ft. Wayne, IN 46805.

Mar 26 Birmingham Microprocessor Group will meet at Southcentral Bell Company headquarters bldg. at 2 P.M. For further details write or call Jim Anderson, 2931 Balmoral Rd., Birmingham, AL 35223; (205) 897-9630.

Mar 28 Computer Amateurs of South Jersey will holds its meeting at the National Park Municipal Bldg., 7 So. Grove Ave., National Park, NJ at 7:30 P.M. For details call (609) 541-1010, or (609) 541-8296.

Mar 29 Ventura County Computer Society will meet at Camarillo Public Library, 3100 Ponderosa Dr., Port Hueneme, CA 93041 at 7:30 P.M. For more information write: VCCS, P.O. Box 525, Port Hueneme, CA 93041.

Mar 30 Space Coast Microcomputer Club will hold its meeting at 7:30 P.M. at the Merritt Island Library, Merritt Is., FL. Contact Ray Lockwood at (305) 452-2159 for details.

Mar 30 Small Computer Engineering Association of Minnesota (SCEAM) will meet at the Resource Access Center, 3010 Fourth Ave. So., Minneapolis, MN 55408 at 7 P.M. For more information write to this address or call (612) 824-6406.

Mar 31 Washington Amateur Computer Society has scheduled its meeting to be held at the Catholic University of America, St. Johns Hall. Located at Michigan and Harewood Aves. in Washington, D.C. Contact Bill Stewart at (202) 722-0210 for club details between the hours of 10 A.M. and 12 P.M.

Mar 31 University of Minnesota Microcomputer Users Group (UMMUG) will hold its meeting at the University of Minnesota, Electrical Eng. Rm. 115 at 7 P.M. For further information write UMMUG, Dept. of Elec. Eng., 123 Church St. S.E., Minneapolis, MN 55455.

CALL FOR ARTICLES

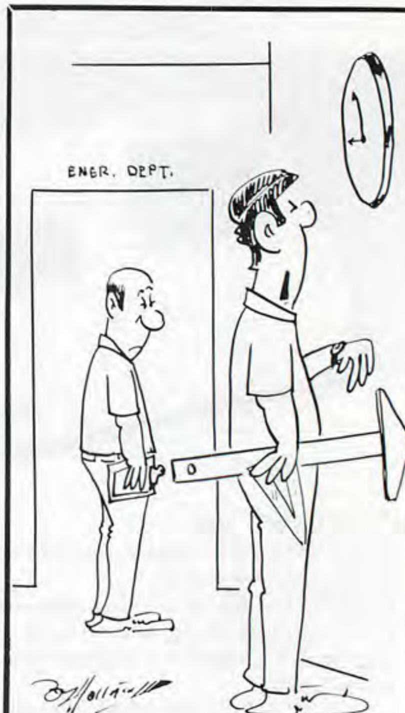
We are actively seeking articles in hardware, software and general applications of microcomputers in industrial, business, science, medicine and personal fields.

Articles authored by individuals during leisure time are remunerated at a rate from $15.00 to $50.00 per published page and articles describing company projects carry author and company byline, but no honorarium is offered. Articles accepted will be acknowledged with a binder check within thirty days of receipt.

Manuscripts should be double-spaced, typewritten pages, one inch margins, and not less than 3½ pages in length (one published page). Pages should be numbered to insure correct text. Photographs should be numbered and labeled on the backside with a description. Tables, listings, etc., shall be on separate sheets. Photos should be taken with uniform lighting and background, in the form of glossy black and white prints. Computer listings shall be printed using a new ribbon to assure darkest print copy. Authors shall supply a statement of their background, expertise and level of accomplishment.

The publisher assumes no responsibility for artwork, photos, models, or manuscripts. Manuscripts are not acknowledged or returned unless accompanied by an addressed, stamped, return envelope.

For article submittal or further information, contact respective editor, INTERFACE AGE Magazine, 16704 Marquardt Avenue, Cerritos, CA 90701 or call (213) 926-9544.



"Twenty-eight thousand, two-hundred thirty-one point seventeen seconds before five PM."

Look To The North Star HORIZON Computer.

HORIZON™— a complete, high-performance microprocessor system with integrated floppy disk memory. HORIZON is attractive, professionally engineered, and ideal for business, educational and personal applications.

To begin programming in extended BASIC, merely add a CRT or hard-copy terminal. HORIZON-1 includes a Z80A processor, 16K RAM, minifloppy™ disk and 12-slot S-100 motherboard with serial terminal interface — all standard equipment.

WHAT ABOUT PERFORMANCE?

The Z80A processor operates at 4MHZ — double the power of the 8080. And our 16K RAM board lets the Z80A *execute at full speed*. HORIZON can load or save a 10K byte disk program in less than 2 seconds. Each diskette can store 90K bytes.

AND SOFTWARE, TOO

HORIZON includes the North Star Disk Operating System and full extended BASIC on diskette ready at power-on. Our BASIC, now in widespread use, has everything desired in a BASIC, including sequential and random disk files, formatted output, a powerful line editor, strings, machine language CALL and more.

EXPAND YOUR HORIZON

Also available—Hardware floating point board (FPB); additional 16K memory boards with parity option. Add a second disk drive and you have HORIZON-2. Economical serial and parallel I/O ports may be installed on the motherboard. Many widely available S-100 bus peripheral boards can be added to HORIZON.

QUALITY AT THE RIGHT PRICE

HORIZON processor board, RAM, FPB and MICRO DISK SYSTEM can be bought separately for either Z80 or 8080 S-100 bus systems.

HORIZON-1 $1599 kit; $1899 assembled.
HORIZON-2 $1999 kit; $2349 assembled.

16K RAM—$399 kit; $459 assembled; Parity option $39 kit; $59 assembled. FPB $259 kit; $359 assembled. Z80 board $199 kit; $259 assembled. Prices subject to change. HORIZON offered in choice of wood or blue metal cover at no extra charge.

Write for free color catalogue or visit your local computer store.

NORTH STAR ★ COMPUTERS
2547 Ninth Street • Berkeley, California 94710 • (415) 549-0858

CIRCLE INQUIRY NO. 33

CIRCLE INQUIRY NO. 1

WHITE COLLAR MICROCOMPUTER

By James S. White

A computer for less than $1000! Can this be more than a toy? Can a low cost microcomputer be a tool useful in practical applications, such as accounting and other types of record keeping, for a professional user who isn't a computer hobbyist?

Small businessmen may be excited by statements that today's microcomputers can have much of the power of the giant computers of twenty years ago—the computers that started the information revolution in large businesses. In more rational moments they may wonder how really useful a microcomputer could be in their businesses, and what kinds of computer systems are needed to handle the jobs these businessmen need done.

Similar questions are raised by people with other widely varying professional data processing applications, such as the operation of professional record keeping services and specialized applications in large businesses.

This month's column reviews some characteristics important in microcomputer (i.e. small, low cost computer) data processing system packages for small volume professional applications. A system package, in this context, is a group of equipment, services, and other products which are combined to form a product which will be a functional tool, able to perform useful work.

Despite the variety of end objectives of their applications, professional data processing systems generally have one major characteristic in common: their purpose is to do a job for a person who doesn't want to know about, or to be involved with a computer any more than to key in the data. The professionals who are computer users are looking for results they can trust, at minimum cost in terms of time and effort.

The results desired by this class of user often are the saving of part of the time and energy otherwise spent for various types of record keeping, calculating, or word processing. The time and energy thus available will generally be used for the professional work (selling clothes, filling teeth, making widgets, or whatever) they can best do and/or most enjoy. Few professional users want to, or can afford to, trade the bookkeeping part of their jobs for computer kit building, programming, and maintenance. Other professionals may desire, as their results from a computer, more (or more accurate or faster) information than they would otherwise have the time to obtain or create.

To obtain these time saving and income-producing results, professional system users are generally willing and able to pay a significant price in money. Rather than choosing the false economy of low cost tools, objective professionals usually try to maximize the dollar benefits to the business or organization, and buy the tools that best do this. A well-designed and implemented computer system will give such users enough time and information that they can pay generously for this equipment and still have considerable net benefits for the business.

For the competent producer of professional computer system packages the result is a market with considerably more profit potential than the hobby computer market.

The characteristics of professional data processing systems can be better understood by comparing them to more commonly available types of microcomputers. The microcomputer systems which have historically (microcomputer history began two or three years ago) been available have been most suitable for design engineers and hobbyists. Engineering product development and training microcomputer systems are clearly different from professional data processing systems in both design and intended use.

Many hobbyists spend as much time as possible building or programming their computers. This apparently typical hobbyist objective of maximizing the time before a system is operational is the exact opposite of professional users' objectives. The hobbyist's objective is often enforced by not knowing for what to use computers once they are operational. Many hobbyists also try to minimize dollar expenses — at any cost.

With such widely divergent objectives, it is a wonder that there is any similarity at all between traditional microcomputer systems and professional data processing systems. Only the tremendous versatility of the computer allows this one type of device, basically the same everywhere, to serve these widely divergent objectives. However, the powerful flexibility of computers tends to obscure the essential differences between systems which are good for professional data processing applications, and systems which are better suited for other types of applications.

READILY FUNCTIONAL HARDWARE

The key characteristic of a data processing system for the typical small business application is that it requires minimal user effort. This practically requires plug-in-and-use hardware. Kit assembly and the matching of components from various sources are clearly not the best use of time for the professional bookkeeper or dentist. Hardware should be functional almost immediately.

Another hardware characteristic important to professional users is sufficient accessible power. Although many hobby computer systems are not marketed in a form suitable for the typical business user, the hardware itself generally has more than enough native capability. The instruction sets, basic hardware operating speeds, possible addressing ranges, and other fundamental characteristics of the basic components are sufficient for the next several years' applications of most users.

Functionality for professional applications requires more than inherent capability. Packaging, in the hardware design sense, is the key consideration. The hardware capabilities must not be buried under the electronic analogs of chrome and gingerbread, inaccessible to the serious user. Also these capabilities must not be so overburdened supporting bells, whistles, and miscellaneous trinkets such as functions for games, that they don't have available the power necessary for hard working applications.

Most streamlined, integrated small computer systems using today's technology provide plenty of power for professional use, although initial purchase costs generally are greater than for do-it-yourself systems.

MAINTENANCE SERVICE

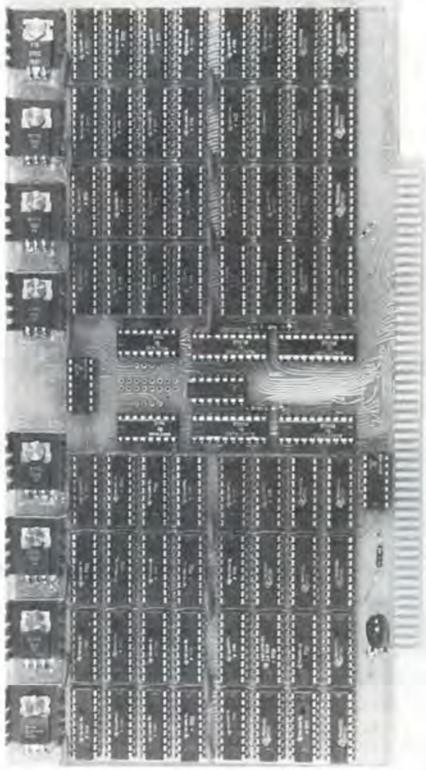
A closely-related requirement is that, after a system has started to work for the professional user, it must remain operational, at least most of the time. However, as

NEW!
32K/16K Static Memory

S-100 BUS COMPATIBLE: Such as ALTAIR™, IMSAI, SOL, POLY, XITAN, etc.

Featuring:

1. **Capacity:** 32768 Bytes for model 3200
   16384 Bytes for model 1600
2. **Addressing:** Each 4k block separately addressable on any 4k boundary. Allows memory to be placed at the top, bottom, and anywhere in between.
   This feature overcomes the problem of working with software that requires memory in different places.
3. **Wait states:** none
4. **Speed:** under 500 ns.
5. **FULLY STATIC:** no clocking, no refreshing.
6. **Memory chips:** High quality/reliability 4kx1, 18 pin static manufactured by Texas Instruments and second sourced by others.
7. **Fully socketed:** even the 16k board contains sockets for all 32k of memory.
8. **Fully assembled,** tested, and burned in.
9. **Lower power** than equivalent capacity of low power 2102 type memory. And only one slot!
10. **DMA Compatible**
11. **Fully Buffered:** All address and data lines buffered with powerful state of the art buffers equipped with Schmitt triggers on their inputs.
12. **Special Thermal Design:** Each 4k of memory has a separate regulator thereby distributing the heat dissipation over 8 separate regulators. They are placed at the top of board to allow the most efficient heat dissipation possible.



Trace Electronics new, flexibly featured, state of the art, 32k STATIC RAM comes fully assembled, thoroughly tested, burned in and warranted for only $999. For further information and fastest service call 215-779-3677.

Call:(215)779-3677 or Use the coupon to order

Assembled, Tested and Burned in: **Model 3200: 32k $999.** **Model 1600*: 16k $599.**

*assembled with sockets for 32k but tested and sold with only 16k.

16k Expansion Kit: Model 1601: 16k of memory to update the Model 1600 to a model 3200. $499.

I would like to order the following:

- ☐ **Model 3200** **32k Static Memory. Assembled and tested $999.**
- ☐ **Model 1600** **16k of memory on the above 32k board. Assembled with sockets for 32k and tested with 16k. $599.**
- ☐ **Model 1601** **16k expansion kit for above. $499.**

My address is:
Name ______
Address ______
City ______ State ______ Zip ______
Phone ______

Enclosed is a check or money order for: ______
Please charge the item(s) I've marked above to my credit card.

☐ Visa Card No. ______
☐ Mastercharge Exp. date ______
4 digits above name (Mastercharge only) ______
Signature ______

Note: Address for credit card orders must be the same as that on file with the credit card company.

TRACE

Mail this coupon to: Trace Electronics Incorporated
P.O. Box 3247 Hampden Station,
Reading, Pennsylvania 19604

do most machines, computers need occasional repairs. Although a hobbyist can return a computer to the store or factory for a month or two, often without hardship, a computer is a tool which is often essential to the work of professional users. If the computer isn't working, neither are the people working, nor are they getting paid. Much worse, customers may be going elsewhere (where the computer is working) and some may never come back. Because most professional computers are more than pastimes or conveniences, prompt and competent maintenance is important to the business of the users.

Microcomputer maintenance service differs from television or typewriter maintenance service in one key regard — good microcomputer maintenance service is not widely available. There are few microcomputer service men or businesses listed in the yellow pages or other professional directories. A surprisingly large number of the stores and other distributors that sell computers don't offer (and can't recommend) a competent repair service. However, good microcomputer repair services are available in many areas, for the users who search them out.

Providing good computer maintenance service isn't easy, even for the maintenance professional. Do-it-yourself maintenance requires too much time and ability to be even marginally realistic for most persons who use their computer in other professions. A competent repair person requires considerable training and experience in a complex, new, and rapidly changing field where very few courses are offered and where any more than a year's experience is exceptional. A competent repair service requires a comprehensive stock of the many components that can fail in a computer system. This stock must be built in a field where some manufacturers can sell all the computers they can produce and thus must choose, in the case of key parts, between selling the parts for repairs, say for $15 each, or as part of a new $1000 system. A competent repair service also requires expensive tools and equipment, although this need might be ignored and compensated for by considerably longer repair work times (paid for by the owners, both in repair charges and in loss of income while their computer can not be used).

A monthly maintenance contract is often the best arrangement for both the vendor and the purchaser of computer repair service because it fixes an amount which both can use as a plan on which to base sound business practices. The price of a typical professional microcomputer maintenance service might be between $100 and $500 per month, depending on the complexity of the system, the durability of the components used, and the quality of the maintenance work. This price generally provides on-site service covering both parts and labor.

Although repair of traditional microcomputer equipment by the typical professional user is completely impractical, some types of business microcomputers minimize the need for professional repair services. A computer system package of this type generally must include:

1) Software, hardware, and/or firmware, with suitable operating instructions, to diagnose easily most of the frequent machine problems. Diagnostic procedures must localize a problem to a plug-in module.
2) A spare parts stock costing little enough to be maintained by each user.
3) A back-up repair service to replenish the parts stock, repair replaced modules, and to repair system problems not covered by diagnostic procedures.

With rare exceptions, equipment of this type is now only available in the medium price ranges (perhaps $10,000 per system). Lower cost systems suitable for do-it-yourself repair by professional users are now being

developed, and may be widely available in a very few years.

Computers can be kept operating, and many types of systems are generally working well with the help of competent and well-equipped maintenance personnel and other maintenance approaches. Potential users can avoid serious problems by considering the extent to which the cash flow of their business will be dependent on the computer, and the consequently needed response time and other maintenance service characteristics.

Another alternative to the previously discussed repair service requirement is to maintain a spare computer system. When this technique is used, replaced inoperable equipment parts are usually returned to the factory or other vendor location for repair. With equipment prices dropping and labor prices rising, this approach is sometimes a practical solution today, and one which is sure to grow in cost-effectiveness and popularity.

SOFTWARE

A computer is what its programs are — this statement is more literally true than most other proverbs. For most business users, the programs for specific business applications are the important software. These programs convert the general purpose hardware into a tool which, with little further guidance, handles the unique needs of a particular user. Application software, more than anything else, determines the characteristics and utility of a system and thus is the major variable determing data processing system performance. Unfortunately, many traditional microcomputers provide absolutely no application software and are therefore of dubious value to the normal business person, at least as marketed.

Because application software is so visible to users, even they with little computing experience can relatively easily evaluate many key points of software suitability and alternatives. The user who understands his busines well (see last month's column for some ideas on this topic), and what functions he wants a computer to perform, can objectively compare how well a proposed computer system (particularly the software) measures up. A computer should do what you expect of it and one can tell whether or not it does. Prospective computers are much easier to evaluate than prospective employees because a computer will always (given proper hardware maintenance) respond identically to identical conditions; there are no high nor low days, no learning nor forgetting. Although a thorough software evaluation study may require several hours or longer, a conclusion can be reached with confidence.

Application software is also the largest variable in system package cost. Programming is an art. A program can be done broad-brush — quickly and cheaply. Or, a program can be written painstakingly, with fine attention to details, special exceptions for the unique using organization, and other factors that will result in easily usable, long lasting software. A high quality software package appropriate for professional use can easily cost several thousand dollars more than the cost of the hardware.

Volume is another key factor in software cost. For example, sometimes a business application is so similar to another application that the same computer software can be used for both. Because software reproduction costs are negligible, the copy of the software can be essentially free. If a business can allocate the cost of its software over several computers, perhaps in branch stores, the cost per computer will be low.

Similarly, a business can pay a low price (and still provide a comfortable profit margin to the competent producer and marketer) by using a software package developed and paid for by another business. Purchase prices also tend to be low for software developed for a general type of business application and sold to many users.

However, the businessman who uses the same software as others in the same field is also running his business the same as his competitors, for better or worse (to the extent that its operation is determined by the computer).

Software is also the largest variable in system cost and performance because programming can be a do-it-yourself activity. Out-of-pocket costs can be negligible if this work is done by staff personnel. Even the total costs of internally developed software can sometimes be quite low, such as in cases where family members do the programming, or where office personnel have been hired for a new business before a full workload exists.

More frequently, employees are paid normal wages and there is no shortage of productive work to do. In these environments, internal development of software can easily result in costs several times greater than external development or purchase.

In any environment, cheap software may be the most expensive. Using software unsuitable for a particular business may require people to do extra work to adapt to the computer, or may produce relatively worthless results. The costs of using unsuitable software may be much more than the cost of purchase or production of properly, specifically designed programs.

Software is generally a key factor distinguishing hobbyist microcomputer systems from commercial systems. Some hobbyist system vendors offer no application software. Most of the hobbyist software available is games, programming languages, or programs for demonstrating business techniques, but not suitable for professional use. Many of today's microcomputer systems would be well suited for business use were software available.

EDUCATION

The use of a new tool, especially one as complex as a computer, requires that a person or business obtain working knowledge. Effective training is required to obtain this understanding. This takes relatively little time and effort of the professional user. The trial-and-error methods commonly used by hobbyists can not be afforded by the user for whom time is money.

In order that a tool's use be accurate and efficient, training should be fairly complete and appropriate to the learners' previous knowledge. A training program should efficiently convey operating knowledge to novices, and subsequently answer specific questions. In a business with employees, an added factor is that good training activities should be motivational, to result in proficiency. Producing a good professional data processing training program is a significant challenge, and one not well met, even to the extent of some obvious needs, by some systems marketed for business use.

Documentation is generally the key training technique for computer systems, primarily because of its relatively low cost for the vendor. An important user benefit of good documentation is that it allows users easily to obtain answers to any of many questions as they arise. Memorizing everything needed to use a computer under all conditions is impossible for most humans.

Documentation is another part of a computer system package which can be rather easily evaluated by novices. One can read the documentation for a computer under consideration, but if the documentation is not understood, it isn't good enough. A better test is to try to operate the computer system, hardware and software, using only the documentation. Generally, a pass or fail mark can easily be given to the system package, and its ease of use clearly seen.

Education is a factor which is not always directly related to system package cost. Whether a system package includes appropriate education for a particular user generally depends on the markets in which the producer and vendor have chosen to support the system. Hence,

this factor often rather clearly distinguishes hobby, engineering, and business computers.

SUMMARY

A data processing system appropriate for professional use has significantly different characteristics than the typical hobby computer. Professionals generally need a complete package they can trust to effectively handle key operations of their business.

The characteristics important to professionals, and whether a given system meets these needs, need not be mysteries delegated to technical specialists. These characteristics can be practically evaluated by an owner or executive who is knowledgeable of his business and the reasons for which he will use a computer. Even as a computing novice, one can make a good start in determining if a system under consideration will be good for the task.

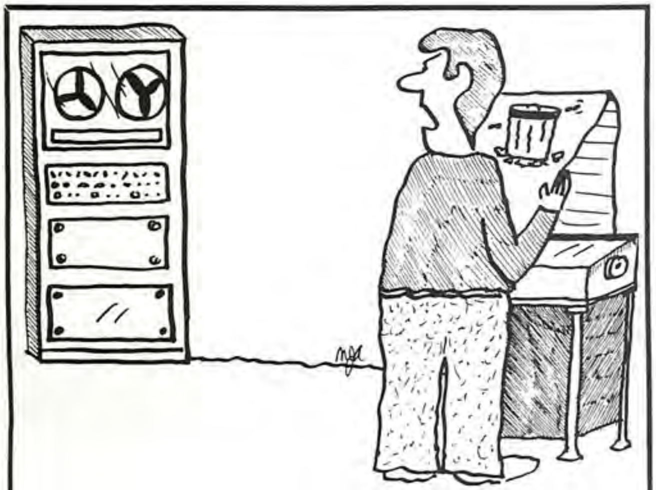
At \$6,000 to \$20,000, the type of data processing system packages presently most suitable for typical professional use cost considerably more than the typical hobbyist system at \$1,000. However, for many professionals, this difference is more than compensated for by increased profits, during very few years. For many of the other prospective users, perhaps a computer is not yet a cost-effective investment.

Some businesses can do useful work by using hobby computers, but most businesses doing so will incur inordinately high total use costs. However, systems with relatively low purchase prices can be useful for specialized applications, or by computer users willing and able to themselves do a considerable amount of the work required to implement a complete system.

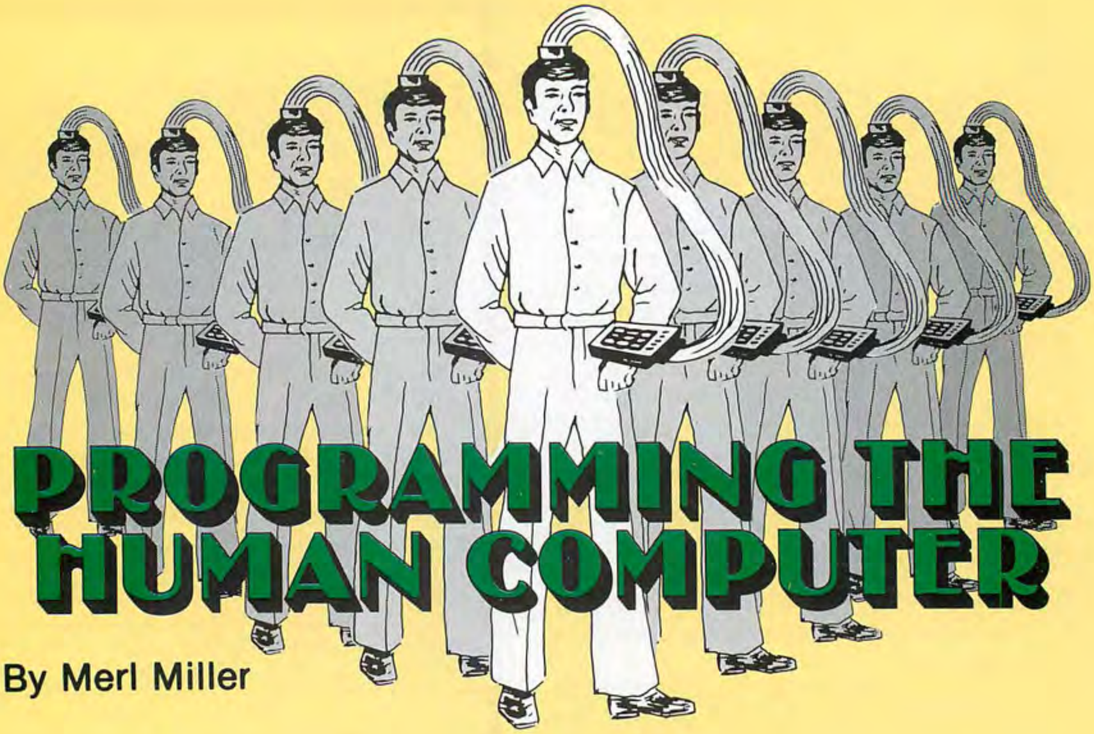
Hardware will be the main topic of next month's column. The main characteristics of a data processing system package not covered this month are the characteristics which result from the various types of computing equipment which can be combined to form an operational package. As was true of the subject discussed this month, the hardware characteristics most suitable for professional use differ from those of the typical hobbyist system.

Specific systems for commercial use will be reviewed in future issues of White Collar Microcomputer. Vendors of such systems are encouraged to send system specifications to this column.

Mr. White will also answer specific user's questions (except will not recommend or evaluate particular brands of equipment or other products). Questions should be addressed to: INTERFACE AGE Magazine, 16704 Marquardt St., Cerritos, CA 90701.



"VERY FUNNY!"



PROGRAMMING THE HUMAN COMPUTER

By Merl Miller

MICROCOMPUTER COURSES

If you want to study microprocessors, where do you go? One place, of course, is your local community college or university. However, at this time, these courses are not fully developed. Educational institutions are slow to react to new things. An instructor has to propose a course to his department head and his dean. If he gets approval from both, he then has to push the course through the bureaucracy before it can appear in the catalog and be offered to students. This process usually takes two or three years. Finally, when the course is offered, there are restrictions on who can take it. What this all means is you probably can't go through the normal university structure to take a microcomputer course. This leaves you with three alternatives, self-study courses, short courses or extension courses.

The university extension course is a good way of learning anything. Generally they are inexpensive and well taught. To find out about these courses call your local community college or university and ask for the Extension Division or Department. If the operator says "huh" ask for the department that teaches off-campus courses. They should be able to tell you who teaches the courses you are looking for.

Probably the best way to tell you about short courses and self-study courses is to give you some examples. Let's start with Logical's Modu-Learn. Utilizing top down design and structured programming the course stresses professional programming techniques. It is designed to teach beginners how to program microcomputers. Although it is intended as a self-study, some computer stores use it for an in-house course. Topics covered are:

Background Material

1. Evaluating Microcomputers
2. Analyzing Software Problems
3. Elementary Programming Techniques I
4. Elementary Programming Techniques II
5. Elementary Programming Techniques III
6. Computer Arithmetic
7. I/O Techniques I
8. I/O Techniques II
9. Higher Level Language Programming
10. Systems Synthesis

The course is 588 pages long and sells for $49.95.

National Semiconductor offers five courses aimed primarily at engineers, but even if you aren't an engineer you may be able to benefit from them. National has had a wide variety of students from secretaries to small business consultants. Course breakdown is as follows:

Course	Length	Tuition
Microprocessor Fundamentals	4½ days	$475
8060 SC/MP Application	4½ days	$475
8900 Pace Application	4½ days	$475
Complex Peripherals	3 days	$395

The courses are expensive and are based on National microprocessors, so you should have a serious application in mind before you consider taking one. However, if you are a small business consultant, a computer store owner or the like, the courses may be ideal. The courses start with basics and cover a lot of good material. For instance, the microprocessor fundamentals course emphasizes basic programming techniques and development tools used in generating applications software. Basic concepts are discussed in class and are then practiced in the lab. Subjects covered are:

- Stored Program Concepts
- A Real Microprocessor
- Microprocessor Development System
- Software Development Tools
- Microprocessor Applications
- Matching a Microprocessor to an Application

The Course is offered at either of the National training centers: Eastern Training Center, National Semi-

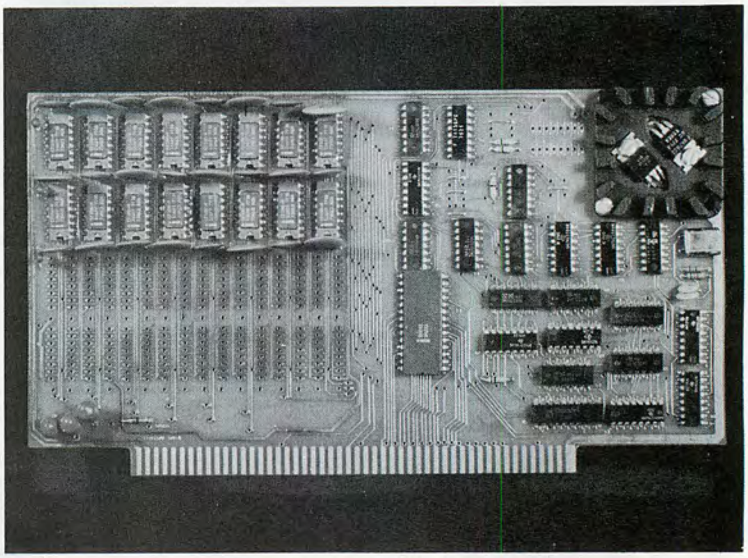
We'd Like To Ask You A Question.

Have you considered the Central Data 16K RAM board? If you haven't, you should. We've combined quality engineering and low prices to come up with a memory board you'll like. Take a look at our features and find out for yourself.

- completely assembled, tested and burned in
- expandable to 32K
- S-100 compatible
- access time of 450ns
- invisible refresh—absolutely no wait states
- power dissipation one-half that of static RAM
- one-year warranty.

You can purchase a 16K RAM board with all these features for only $289. The cost of adding 16K to your present Central Data 16K board is $200. A 32K RAM board, assembled, tested and burned-in, is only $475.

With all those features, don't you think you should be considering ours?



Central data

PO Box 2484, Station A
Champaign, IL 61820

Please send me the following:
☐ 16K RAM board ($289)
☐ 32K RAM board ($475)
☐ additional information
I enclose $_____________.
Ill. residents add sales tax.
Central Data P.O. Box 2484, Sta. A Champaign, Il 61820

Name ______________________
Address ______________________
City ______________________
State ______________ Zip ______________

CIRCLE INQUIRY NO. 4

conductor Corporation, 1 De Angelo Drive, Suite H, Bedford, MA 01730, or Western Training Center, National Semiconductor Corporation, 1333 Lawrence Expressway, Santa Clara, CA 95051. The course is offered free to university professors who are teaching 8060, SC/MP or 8900.

Speaking of university professors, Jim Melsa is chairman of Electrical Engineering at Notre Dame and Dave Cohn is a professor in the same department. They teach a short course on microprocessors for professional societies and businesses. The course is specifically designed for people who need an understanding of microprocessor technique and application. Cohn and Melsa have successfully taught this course to such diverse groups as NEDA (National Electronic Dealers Association) and ISA (Instrument Society of America). The course does not assume any prior background in computers or microprocessors. It covers the following:

Machine Structure
Stored Programs
Jump Instructions
Use of Memory for Data Storage
System Monitor
Use of Terminal for I/O
Editor
Symbolic Assemblers
Stacks and Subroutine
Microcomputer Architecture
Interface Devices
Interrupts and Real Time Clocks
Peripheral Equipment
Comparison of Various Processors
Other Approaches to Programming

This gives a little information on short courses. However, these are not the only courses available. In fact, this is a fairly small sample. Courses are also taught by people such as Wintek, Adam Osborne, Integrated Computer Systems, Sybex and others. If you are looking for a small at-home business using your microcomputer there is an opportunity here. A central list of who offers what course is not available. If you were to start such a list and charge people for being listed, you might be able to make some money at it.

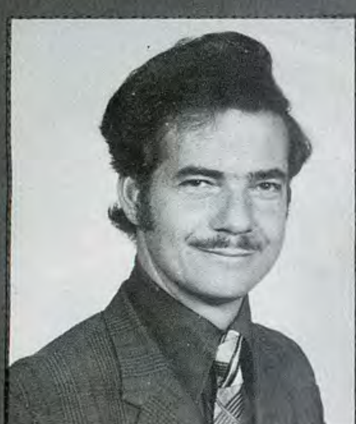
I can get you started by giving you the names of the three I mentioned: National Semiconductor, write the Training Center nearest you; Logical Services, Inc., 711 Stierlin Rd., Mt. View, CA 94043; or Dr. James Melsa, Chairman, Dept. of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556.

If you know of a course, you can send me some information on it. I can't promise to do anything with it at this time, but I will keep a file. My address is: dilithium Press, 30 N.W. 23rd Place, Portland, Oregon 97210.

"The note says he's from 'Computer Dating'."

... FROM THE FOUNTAINHEAD

By Adam Osborne



For all of you microcomputer owners who want a 16-bit CPU that uses your existing hardware, Fairchild may have just the product. Their new 9440 microprocessor is a single-chip CPU that has the instruction set and execution speed of a Nova 1200 minicomputer. Fairchild will be selling a three-card system that will fit into any standard S-100 bus microcomputer frame. The three cards will provide a CPU with interrupt logic and direct memory access, plus 16K bytes of *read/write* memory. I expect the retail price for the three cards will run somewhere between $1,500 and $2,000. O.K., if that is too steep you can always stay with 8 bits.

In case you don't like the Nova instruction set or the Fairchild 9440 prices, you might consider the TMS 9900 systems being offered today or the Alpha Micro System, which has an instruction set remarkably like the LSI-11. But do not hold your breath waiting for the Intel 8086 or the Zilog Z8000. I doubt if you will see either of these products before September of 1978; and they will probably not show up in hobby computer configurations until early 1979. Even then they are going to make current 16-bit systems look inexpensive.

There is a rumor running around that John Draper is designing new telephone communications equipment for Apple Computer Corporation. Now, that is a dynamite combination. John Draper may be better known to some of you as Captain Crunch. John probably knows more about the world's telephone systems than anyone in the world (a fact that has caused great sorrow and consternation to more than one telephone company official). Let us all wait to see what John produces for Apple Corporation, because if anyone can generate a terrific telecommunications product for the computer hobbyists, John Draper can.

Indeed, all you telephoners, TDL is not one of the lily-white companies, as I reported in my September column. I have had a number of people write to complain about TDL's product delivery policies, particularly with regard to software deliveries, and that a lady at customer service gets very nasty on the telephone when people call to complain. But let's be fair. That does not make TDL one of the bad guys, *it simply puts them in there with the pack.*

I still maintain that if you want to get what you pay for, at around the time you pay for it, you should order products cash on delivery, or should pay only for what you have seen operating. Buy products not promises. When you buy promises the chances are that whoever made you the promise will cash your check and then try to fulfill his commitment.

E&L Instruments and Bill Godbout, about whom I have said nice things in the past, continue to receive praise. I have received no negative inputs regarding E&L Instruments and just two regarding Bill Godbout; that must be classified at the noise level. I would like to add two new names to the good guys list:

Seals Electronics, Inc.
P.O. Box 11651
Knoxville, TN 37919

SD Sales Company
P.O. Box 28810
Dallas, TX 75228

Readers of my column have gone out of their way to write and telephone praising the above two companies. These two companies will not cash your check unless they have the product you order, they have fast turnaround, and whatever they sell works. Now, those are impressive statements that apply to very few manufacturers in the microcomputer industry.

A number of good things are happening in the area of published software. In addition to the books of software available from SRI and my own company, dilithium Press is planning some very interesting books of business software to be published during 1978. Having talked to the people at dilithium Press about their plans, I encourage you to keep watching their ads. To tie it all together, Jim Schreier is putting out the *Schreier Index to Published Microcomputer Software.* Now here is a long overdue product. Jim can be contacted at:

The Schreier Software Index
4327 East Grove Street
Phoenix, Arizona 85040

I don't see how anyone with access to a microcomputer can avoid getting the SSI Index.

There is also a proliferation of low-cost business software packages that are quite limited in their overall capabilities, but that will work within the microcomputer hardware configurations that you can buy today. One interesting low-cost set of programs is offered by the Structured Systems Group. They may be reached at (415) 547-1567.

Turning to hardware for the moment, Advanced Micro Devices' arithmetic processor (the Am9511) can now be bought. For months they have been telling us to wait another month; so in case you think it is just the microcomputer manufacturers who play those games, you are wrong — the semiconductor manufacturers are just as guilty. Right

Branched to Page 49

SENSE LINE

By Gary Young

SO YOU WANT TO BE A COMPUTER STAR!

You've finally built your system, mastered its language, and you now spend every waking hour sweating over a hot terminal. Your job at the corner drugstore is boring and you spend your working hours of thinking of programs you'll write when you get home at 5. You have finally decided it's time for a career change to the glamorous world of computer professionals. After all, the money is good, plenty of jobs available, and you enjoy the work anyway, right?

In what area are you going to specialize — programming, analysis, operations, or maintenance? What do you enjoy doing the most and feel the most competent at on your own system — writing original programs? thinking up original programs? implementing someone else's programs from magazines? operating the system? or building and maintaining it? Chances are you prefer doing a combination of these things. Say you lean more toward software and less toward the hardware. Look at what the professional world has to offer.

In what kind of software are you most interested? Operating systems? scientific or engineering programs? business data processing applications? or computer control applications? The system programmer is half man/half machine. He speaks in hexadecimal and thinks in logical binary. If he works at an IBM installation, his tasks may be limited to wearing blue shirts and installing IBM-written modifications. However, whether on IBM or more flexible systems, he is considered the High Priest to the Digital God. The scientific or engineering programmer has a heavy math and science background. He speaks in symbols, and can look at a FORTRAN program and predict the weather. The business application programmer is the one responsible for errors in your bill, delays in your paycheck, and printing more paper than imaginable. He is also the most in demand. There are job openings in industry. The computer control programmer is a new breed. He was a system programmer, but now, with the advent of controlling minicomputers, has a group of his own. His efforts monitor power plants and refineries, control factories, and delay traffic signals.

The type of commercial computer installation will range between two extremes. On the small end there is the Ma and Pa mini installation that is looking for one person who can be DP manager, system programmer, applications programmer, operator, and data entry clerk. On the other extreme there is the large corporation with twin IBM 370/168 computers, 16 megabytes of memory, and a thousand programmers, analysts, and operators tending the system like a queen in a beehive. They are looking for a COBOL application programmer to maintain the coffee break timekeeping system for third floor employees. The computer is in a different city, you get one run a week, and the project deadline is in three days. Obviously the average environment is somewhere in the middle of these extremes. Two years ago the average could have been toward the larger systems, but the growth of mini and micro systems has opened up many smaller shops.

CIRCLE INQUIRY NO. 11

The salary in the computer profession is excellent, but don't be influenced by high raw salaries. What is high for the Midwest will be average for the coasts, and so are living expenses. Beware of data processing employment agencies. They advertise high salaries in the newspaper, but no one is really going to pay $35,000 for a keypunch operator! The newspaper is full of want ads for programmers and the salaries are high. What's the catch? Many of the ads are by employment agencies, each advertising the same job or recruiting bodies to try to make jobs with an employer. The salaries listed are usually the maximum an employer will pay if he finds someone perfect for the position, not at what he may start you. The big money usually comes from the bigger companies, the ones looking for a maintenance programmer. That company has to pay a higher salary to keep people because of high turnover. They want to hire only someone with years of COBOL experience in their particular application. The fallacy is that the company just lost its perfect programmer because he got tired of doing just COBOL on their application. Then that programmer is faced with a job market that wants to hire him to do the same thing he just quit to keep from doing — more COBOL.

For these reasons, the average programmer lasts at a job only about 19 months. This makes programming one of the highest occupations for people making total career changes by age 35. A programmer simply gets bored! He is seldom doing the variety of tasks which he could be doing at home on his own system. Usually he is removed from the computer and his work, buried in non-programming paperwork, and under pressure. Grim? Only if you were impressed by the commercials for data processing schools.

There are still plenty of good programming and DP-related jobs, but they must be searched for carefully. The personal computer owner will not be merely an average programmer. Computers are his lifestyle, not just a job, and he is not easily satisfied by any mundane programming tasks. He needs a position that will challenge him beyond that which he can find at home.

If you are considering moving to the professional computer world, by all means try it. The industry needs talented devotees. Give considerable thought, though to exactly what you want to do and in what kind of environment you want to do it. Use caution as your fingers walk through the the want ads, and prepare yourself for a more pressured, structured, and formal environment than found in your home den.

FOUNTAINHEAD

Vectored from Page 46

now the Am9511 is without a doubt the best arithmetic processor on the market. The calculator chips being sold as arithmetic processors are no match for the Am9511, and you should not consider them in competition. But then, unless you need logs, exponentials and trigonometric functions, you do not need the Am9511.

Those of you who are becoming more sophisticated and are working with hardware, you may want to consider taking a course that would give you some hands-on hardware experience. If your local university has a reasonable course, that is the place to begin because it is likely to be inexpensive. If you go commercial you will pay more money, but good courses are available. For starters try Integrated Computer Systems (213-559-9265) or Tycon (703-951-7030). Tycon are the Bug Book authors and Integrated Computer Systems is one of the most reputable (and possibly the oldest) companies in the course and seminar business.

EUROPEAN

INTERFACE

By Hans Drewitz and Roland Hesse

Paris is one of the more interesting places of this world and not only because of the Louvre or the great French food. Located in the middle of western Europe it concentrates European business activities as very few other cities do. Large international companies ran their European operations out of Paris and the French industry is to a very large extent controlled by headquarters centralised in Paris. With roughly 10 million people it represents itself as a great opportunity for end-user-oriented business.

This is the place from where we will regularly report for you in this column. From here we will give you a life report on the arrival of the Microcomputer Revolution in Paris, in France and in Europe. It has not really hit Europe yet. Events like the 1976 Personal Computing fair in Atlantic City are still to come.

However, there is sufficient evidence that Europe will also in this matter follow the footsteps of North America.

Computer clubs are beginning to become active and some computer stores are selling microcomputers, peripheral equipment and software over the counter.

In one of the next issues we plan to write a report about the few pioneer computer stores in Europe, whose owners or managers, with a few exceptions, we know personally.

We will try to cover the major European exhibitions and report on European-developed products in the microcomputer area.

We also want to point out some of the unique requirements in our area. Differences between Europe and the USA which can and often do cause misunderstandings and headaches for hobbyists and computer shops in Europe and manufacturers and suppliers on the U.S. side of the ocean. For example, some of our correspondence with U.S. companies takes months to reach us simply because somebody forgot to mark the letter "AIRMAIL." Imagine the frustration if you wait for a page of a badly needed engineering change documentation or for a replacement or a burned-out IC.

By informing about requirements and special condi-

INTERFACE AGE
BACK ISSUES
Available in Limited Quantities

Vol. 1, Issue 5, APRIL, 1976
Introduction to Microprocessor Technology; Bubble Memories Are Coming; Calculatin' Engines; Teleprinter Maintenance; Altair Alterations.

Vol. 1, Issue 11, OCTOBER, 1976
National's New Portable Terminal; SA-400 Mini-floppy; CSC—Experimentor 300/600; Software Power for Your 6800; ESP-1 Software Package

Vol. 1, Issue 12, NOVEMBER, 1976
Build a Simple A to D; Super Chip FD1771; RCA ASCII Keyboard Modifications; Protecting Stored Programs; New Product Guide; SWTPC 6800 ROBIT-1

Vol. 2, Issue 2, JANUARY, 1977
A CRT Terminal Using the M6800 Chip Family; AMI's EVK Series Microcomputer; Build a 3-Digit A/D; Microcomputer Design Aids; Cromemco Bytesaver™

Vol. 2, Issue 3, FEBRUARY, 1977
8080 Octal Monitor Program; Legion: An Experiment in Artificial Intelligence; Microcomputer Stock Options; Building a 12-Bit A to D Converter

Vol. 2, Issue 4, MARCH, 1977
Menace of the Micro World; New Product Guide; The Qube; Card of the Month—Cromemco T.V. Dazzler™; Z-80 MITS 12K Extended BASIC Patches.

Vol. 2, Issue 6, MAY, 1977
Computrac 2000; The Floppy-ROM™ Experiment; Robert Uiterwyk's 4K BASIC Interpreter Program; Help Your Computer Understand Your Voice

Vol. 2, Issue 8, JULY, 1977
Diablo*Output Driver Routine; Some Further Notes on Robert Uiterwyk's Floppy ROM 4K BASIC; Microcomputerized Combination Lock; PIA Test-IOTST

Vol. 2, Issue 9, AUGUST, 1977
The Shadow of What?; A KIM-1 Sidereal/Solar Clock; Solar Eclipse Prediction by Microcomputer; Viking UPLINK/DOWNLINK; Star-Ship Simulation-Part I

Vol. 2, Issue 10, SEPTEMBER, 1977
General Ledger Program; Microcomputers: The Intelligent Terminals; Star-Ship Simulation-Part II; PerSci Intelligent Floppy Disc Controller

Vol. 2, Issue 11, OCTOBER, 1977
Computers for Survival; Portable Automated Mesonet; The Energy/Environment Simulators; Biorhythm; Assembly Language Structured Programming; Video Chase for 8080/VDM

Vol. 2, Issue 12, NOVEMBER, 1977
Point Humans: Optical Perception of People and Computers; A Byte of Music; General Ledger Package: The Micro Bookmaker, Conclusion; Number Base Conversion Program

Vol. 2, Issue 13, DECEMBER, 1977
Household Finance System I and II; Personal Accounts Payable Program; The Pocket Computer; A Firmware Model Railroad Controller; Up Your Terminal; Multvibrators; Piranha

Vol. 3, Issue 1, JANUARY, 1978
The Word Processor; Medical Accounts Receivable Package; How to Buy an Apartment House; Turning Toward Meca; Computer(ese) Speech Writer; Tax Calculation Program

INTERFACE AGE Magazine Dept. BI · P.O. Box 1234, Cerritos, CA 90701

Name (Print) ________ Address ________ City ________ State ________ Zip ________

Please send me:

Issue	Qty	Price	Total	Issue	Qty	Price	Total	Issue	Qty	Price	Total
APRIL 1976		2.25*		MARCH 1977		2.25*		SEPTEMBER 1977		2.50*	
OCTOBER 1976		2.25*		MAY 1977		2.25*		OCTOBER 1977		2.50*	
NOVEMBER 1976		2.25*		JULY 1977		2.50*		NOVEMBER 1977		2.50*	
JANUARY 1977		2.25*		AUGUST 1977		2.50*		JANUARY 1978		2.50*	
FEBRUARY 1977		2.25*									

*Price includes 50¢ for postage and handling.

TOTAL ENCLOSED $ ________

☐ VISA ________ ☐ master charge ________ Exp. Date ________ Sig. ________

You may photocopy this page if you wish to keep your INTERFACE AGE intact. Please allow six weeks for delivery.

tions on the European side, we believe we can make a contribution to bringing the European amateur and spare time professional closer together with the engineers and businessmen in Silicon Valley.

Since Europe is very heterogeneous and two persons can cover more than one, we have decided to do this column together. This should also give you a more balanced picture since we both have to agree on what we write.

If you out there in Europe would like us to write about your product, service, computer store, computer club or experience with your American supplier, call us on Paris 333-18-14 or 962-23-97. We will be glad to discuss any item with you.

The following material is now somewhat dated since the magazine elected to hold Hans' and Roland's column for inclusion into the INTERNATIONAL SPECIAL. The report, however, contains information of current value. —Editor

One of the biggest and best known international exhibitions took place in Paris from September 21 through September 30 for the 28th time. This is the "SICOB-Salon International de l'Informatique, de la Communication et de l'Organisation du Bureau." As the name says this covers all kinds of data processing, communications and office organisation equipment. There is, however, a certain concentration on small and medium size data processing equipment. For this reason this exhibition should be of particular interest for all manufacturers of microcomputers, peripherals and associated products.

The SICOB usually attracts about 300,000 visitors from over 100 countries. In 1976 one thousand seven hundred eleven trade marks from 28 countries were exposed on 600 booths.

Since a few years the exhibition takes place in a special building in one of the suburbs of Paris. Outstanding facilities, excellent organisation and easy commuting are strong points of the SICOB.

Unfortunately the exhibition area of 894,000 square feet cannot be extended and exhibition space is difficult to obtain. Space reservation for 1978 starts right after SICOB 1977 closes its doors. Anybody who wants further information contact: French Chamber of Commerce in the United States, 1350 Avenue of the Americas, New York, NY 10019, Tel. (212) 582-4860; or contact Convention Informatique, 6, Place de Valois, 75001 Paris/France, Tel. 261-52-42.

Most of the large U.S. and European manufacturers of data processing equipment for the small business market are present. This year we noticed for the first time some of the U.S. microcomputers. We saw Commodore's PET with delivery vaguely quoted to be in third quarter 1978.

A supplier of storage media used an IMSAI system

"And for obvious reasons, we've dedicated these terminals for synchronous communication."

and demonstration software from Euro Computer Shop. We also saw an OSI Challenger. French microcomputers are also coming into the picture.

R2E presented their "MICRAL" product line which is in the upper range and more Mini than Micro. R2E also exhibited at the latest NCC in Dallas.

Fontel Informatique and MBC, which we had seen already in '76 have made significant progress with their products.

Fontel's system, which they sell to OEM only, is available in 8080 or Z-80 version.

MBC's "ALCYANE" is an 8080-based development. Both systems provide the standard interfaces and attachment of floppy discs but are using their own bus architecture. Unfortunately they are not S-100 bus compatible and their users can not take advantage of the great variety of S-100 bus boards.

Olivetti presented its P60XX product line with the P6040 competing in the low cost microcomputer market. For approximately $3,500 it offers 16K ROM, 3K RAM, an alphanumeric keyboard, a 16-column printer and a special small 3K diskette, all in one frame.

During the SICOB, seminars, forums and conferences take place. Most interestingly this year for the first time a session on "Personal Computing" was on the program. It was organized by an association of young executives. At the conference table were three members of the Association of Young Executives, representatives of the Ministry of Industry, a French manufacturer of microcomputers (R2E), a representative of a computer store (Euro Computer Shop) and a private user.

A large number of people attended this conference and participated very actively in the discussions. The conference covered the technology development of computers up to today's microcomputers and the impact these developments could have on our personal life. Two items are worth noting, a general complaint that most of the literature and instructions are only available in English; there was a general feeling that people are not as familiar with computers as in the United States and thus the market for home computers will take longer to develop.

Of special interest were the remarks made by the representative of the Ministry of Industry who, of course, would like to see more French manufacturers to get into this market. He listed six conditions for a successful development in France:

1) Good understanding and sizing of the professional and home computer market potential;
2) Support of French component manufacturers;
3) Development of software;
4) Distribution network;
5) Improved methods of production;
6) The determination and willingness of companies to invest in this market.

With this conference, SICOB has recognized the era of Personal Computing and we would not be surprised to see a larger number of exhibitors in this field in next year's exhibition.

***Hans Drewitz** has a background in economics and has been working with computers since 1960. He has held jobs as a programmer, systems analyst, systems programmer and various management positions. Presently he is working in a key product management job covering Europe and some other countries.*

***Roland Hesse** has a Masters Degree in engineering from TH Dresden, Germany. He has been working in the computer field since 1961 as a programmer, analyst and in various management positions in Germany, the USA and France.*

Both Roland and Hans have been among the pioneer European computer hobbyists.

UP AND RUNNING

TDL EQUIPMENT USED BY NEW JERSEY PUBLIC TELEVISION TO PROCESS NEW JERSEY GUBERNATORIAL PRIMARY ELECTION RETURNS

John Montagna, computer engineer (above left), lead this successful network team in generating election results speedily, efficiently and reliably using predominantly TDL hardware and software. Montagna created three programs to get the job done. The text for a SWAPPER program was written and assembled using the TDL TEXT EDITOR and Z80 RELOCATING MACRO ASSEMBLER. The SWAPPER text and all debugging was run through TDL's ZAPPLE MONITOR. The relocatable object code was punched onto paper tape. A MAIN USERS program updated votes and controlled air display. An ALTERNATE USERS program got hard copy out and votes in. The latter two programs were written in BASIC. Montagna modified the ZAPPLE BASIC to permit timesharing between the two USERS programs.

Four screens were incorporated, two terminals entered votes as they came in and were used to call back votes to check accuracy. Montagna called on the power and flexibility offered by TDL's ZPU board and three Z-16 Memory boards.

Montagna's setup worked constantly for over four hours updating and displaying state-wide and county-wide results without flaw.

"I chose TDL because they have all the software to support their hardware, and it's good; it has the flexibility to do the job."

John Montagna

We salute John Montagna and NEW JERSEY PUBLIC BROADCASTING for spearheading the micro-computer revolution.

TDL's XITAN SYSTEMS have the capacity to do similar tasks for you. Write to us for XITAN information and the name of your nearest TDL dealer.

TECHNICAL DESIGN LABS

RESEARCH PARK BLDG. H 1101 STATE ROAD
PRINCETON, NEW JERSEY 08540 (609) 921-0321

SFS WALLETSIZE SPACESHIP SIMULATOR

PART I OF 2 PARTS

By Lance E. Strickler, John H. Buffington, III, Ivy Fleck Strickler

INTRODUCTION

We were a small, Philadelphia-based group of technology freaks—you know the kind, always building a better light organ or a home music synthesizer. We had gone through a series of calculators, working our way up from a simple number-cruncher to the more sophisticated card programmables, and finally became the first kids on the block to build a microcomputer kit. By adding our previous experience in audio electronics and stock market computers, we managed to get the thing to sing and dance, so it was time to go on to something more challenging.

THE SPACE RACE SYNDROME

We have always been interested in space, from the very first Sputnik right up to the present Space Shuttle. Unfortunately, even with the space program growing by leaps and bounds, there is still little room in the industry for the stay-at-home experimenter. You simply can't build your own spaceship and take off. We have tried everything from a cardboard carton "kiddieship" built for a younger brother to a more sophisticated "fantasy booth" made of styrofoam and cardboard. Knobs were glued onto the foam and meters were rigged out of photographs and drafting tape. On black and white videotape, the results were impressive enough to give us the courage to go on.

Recently we took part in a course called "Orientation to Flight." The course gave us flight theory and several hours on both hydraulic and electromechanical flight simulators (Photo 2), plus some brief "in flight" training. The simulators provided an eye-opening experience. The hydraulic simulator, seen from the ground as almost an amusement park ride, suddenly became real as you were strapped inside. Although the simulation was a limited one, so much direct and peripheral attention was taken up while monitoring the instruments and allowing the mind to experience the "flight" that the surrounding walls were forgotten. We were flying!

Then it occurred to us: here was a way we could sail through the solar system. Here was a chance to learn about computers, about physics, about space flight, without a huge expenditure of money and time. Here was an opportunity to combine all we had learned through years of model rocket launches (Photo 3), and two decades of varied electronic experience.

The "SFS Walletsize" was about to be committed to paper.

Nearly everyone reading this article has at one time played some version of "Space War" or "Lunar Lander." I usually jump at the chance to play a new version of any space game, not just for the fun of it but also to evaluate how real the simulation feels. Unfortunately, most of these games were written to run on a time-shared printing terminal, so the end result is often less than thrilling. Two dimensional Space War, even when displayed on a CRT, became boring, and 2-D lunar lander, with keyboard input, is even less enjoyable. After all, I can play 3-dimensional advanced Star Trek complete with long range tracking, course controls, sensors, shields, phasers, photon torpedoes, Klingons, Romulans, Vallicians, Nubian freighters, a starbase with various commands and even tractor/transporter beams, on an HP-67 or -97, using only six program cards. Surely something more spectacular could be done using a microprocessor.

THE BEST LAID PLANS . . .

The simulator project was approached as a sort of "team hobby." This was to be a fun project, something in which we could all get involved, and would eventually all learn to fly. We did, however, feel it was best to set out a list of objectives for the project, and also to keep minimal records on the amount of time and money spent. A bill of materials on the cost is shown in Table III at the end of this article, for those who want to follow in our footsteps.

Our first objective became **realism**. We would make the simulation so elaborate that all of the pilot's attention would be absorbed and the real world ignored. We also relied on distraction, adding six small speakers to provide sound cues, and three video monitors (one a 25" diagonal screen to substitute for a window), as well as a host of controls, instruments and indicators, all either connected to the processor, or intertwined with "pinball machine logic." Thus far, no one has noticed the lack of weightlessness nor the missing crush of acceleration.

A second vital factor was **cost** (see Tables II and III). Since this was a spare time project, all expenses would be out of pocket, and had to be cut to the minimum. We decided on a final budget of $500, a figure which seemed to hold. The rule of thumb became: If it's not free, scrap, surplus, junk or from our inventory (in that order), then fake it. If you can't fake it, then, and only then, buy it — preferably at a discount. We felt our latent scrounging instincts coming to the fore.

A third objective was **software simplicity**. We decided to simplify the software requirements by writing most of the software in BASIC. However, in order to support the illusion of a real-time simulation, some video display functions were allocated to machine language subroutines which would be called from BASIC. At this point, the advantages of running 8080 BASIC in a Z-80 CPU and the marvels of the Z-80 itself became apparent. First of all, having a second set of registers in the Z-80 is like having a second CPU with which to work when you interrupt BASIC. The advantages of block instructions became obvious when we had to move screenfuls of video. This CPU also enabled us to use Zilog CTC programmable counter-timers and a handful of other parts to substitute for software-intensive analog-to-digital (joystick, thrusters), and digital-to-frequency (speakers) conversion. With those time wasters out of the way, BASIC was free to do what it does best: solve equations, handle strings and pass data to both machine language subroutines and peripherals. As if this were not enough, BASIC also has command over a background real-time control structure.

A fourth objective was **adaptability** of the craft. We wanted to be able, by changing the BASIC program, to simulate an Apollo capsule, a Lunar Lander (LEM), a

PHOTO 2 Author (Lance E. Strickler) piloting craft during research.

space shuttle or any one of a number of science-fiction vehicles (see Table I). After a bit of research into the problem, we decided to list the different simulation programs we felt confident we could write (given infinite time). From this list we then determined what controls and displays were necessary to operate these various spacecraft and their simulations. As it turned out, one 3-axis (roll, pitch and yaw) joystick, three thruster translation switches (up/down, port/starboard, forward/aft), three linear action main thrust controls, six assorted switches and ten lamps would serve as a good average for active controls (see Photos 4 and 5). However, since the I/O interface is so simple and cheap, we have added the capability of controlling 64 of the indicator lights and reading 64 of the switches. Some of the logic is hard-wired into the switches so that one bit on the input port is reading a group of switches. This also eliminated the need for an input mux and decoding software. For example, when the pilot takes over manual control of the main engines, he must throw the auto/manual switch to "manual," select orbital or main engine switches, and set the electrical subsystems from "off" to "primary" or "secondary." He then moves the safe/arm switches to "arm," fires the engine and optionally arms and fires explosive bolts for lift-off, etc. This uses up twenty surplus switches, but when hard-wired uses only eight input bits (see Figure 1). "Pinball machine logic" of this sort is located in many of our general spaceship systems as well as in "real" aircraft and spaceship simulators.

A fifth objective was that the **versatility** of our microprocessor not be impaired. At first we discussed using a collection of Sphere's, SC/MPs and so forth which we had on hand, rather than our main processor. This would have meant using the Spheres to run the screens, and the SC/MPs to run the switches and lights. A fancy setup, but what a mess! We finally decided that we would use one CPU to run the whole show, adding two more video cards (one purchased, one borrowed), to the one

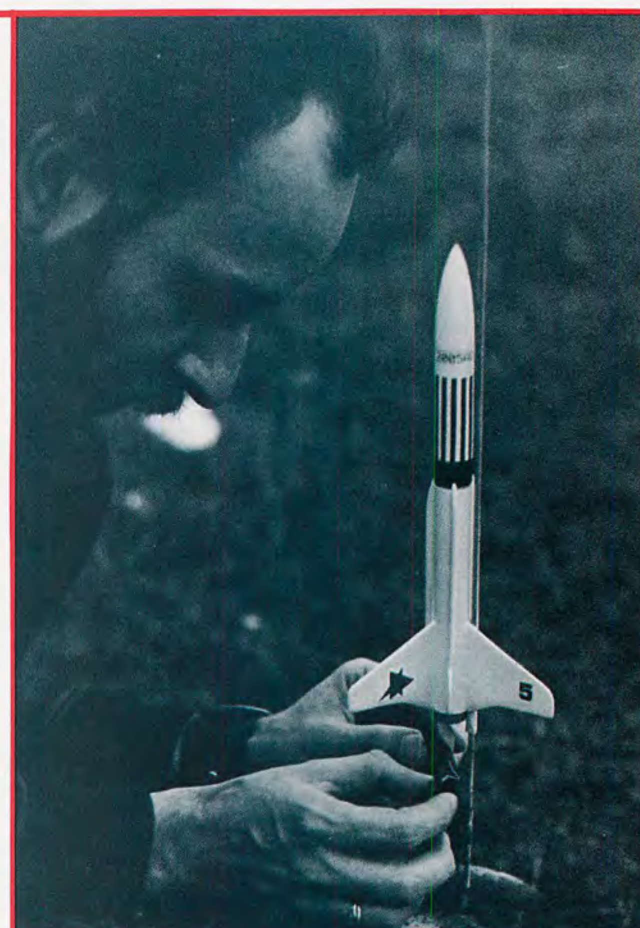


PHOTO 3 Model rocket being prepared for launch.

- I. Apollo Program Simulation
  - A. Saturn 1B Launch Vehicle
    - 1. Earch Orbit
      - a. Insertion
      - b. Circularization
    - 2. Skylab Docking
    - 3. Apollo Rendezvous
    - 4. Soyuz Docking
    - 5. Controlled Re-entry
  - B. Saturn V Launch Vehicle
    - 1. Earth Orbit
      - a. Launch
      - b. Insertion
    - 2. Circumlunar Track
      - a. Translunar Injection
      - b. LEM Docking
      - c. Mid-course Corrections
        - 1. 5 hour, 6 minute
        - 2. 55 hour, 30 minute
        - 3. 63 hour, 15 minute
      - D. Lunar Orbit
        - 1. Insertion
        - 2. Circularization
        - 3. LEM Lunar Landing
        - 4. LEM Ascent, Docking
      - E. Transearth Injection
      - F. Midcourse Correction
    - 3. Controlled Re-entry
- II. Space Shuttle Simulation
  - A. Earth Orbit Insertion
  - B. Orbital Operations
    - 1. Orbit Circularization
    - 2. Satellite Mission
      - a. Orbital Release
      - b. Repair Rendezvous
      - c. Deep Space Probe Launch
    - 3. Shuttle Rendezvous and Docking
    - 4. Skylab Docking
    - 5. Research Mission
      - a. Upper Atmosphere
      - b. Solar Phenomena
      - c. Astronomical
    - 6. Cargo Delivery
  - C. Re-entry, Glide Landing
- III. Science Fiction Simulations
  - A. Extended Space Shuttle
    - 1. Asteroid Pick-up/Mining
    - 2. Venus Fly-by
    - 3. Mars Exploration
      - a. Lander Release
      - b. Lander Orbit/Docking
    - 4. L5 Shuttle Service
    - 5. Lunar Shuttle Service
    - 6. Comet Chase
    - 7. Orbital Trash Pick-up
  - B. Interplanetary Spaceliner
    - 1. Space Station Docking
    - 2. Planetary Cruise
  - C. Interstellar Research Vehicle
    - 1. Hyperspace Jumps
    - 2. Near Star System Exploration
      - a. Star Orbital Operations
      - b. Planetary Operations
    - 3. Binary Star System Exploration
    - 4. Gas Cloud Sampling
    - 5. Mapping Expedition
  - D. Earth Orbit Tactical Ship
    - 1. Offensive Missile Evasion
    - 2. Satellite Search and Destroy
    - 3. Alien Orbital Dogfight
  - E. Starship Enterprise
    - 1. Warp Drive Voyage
    - 2. Klingon Battle
      - a. Surprise Attack
      - b. Search and Destroy

Table 1. Simulation Outline

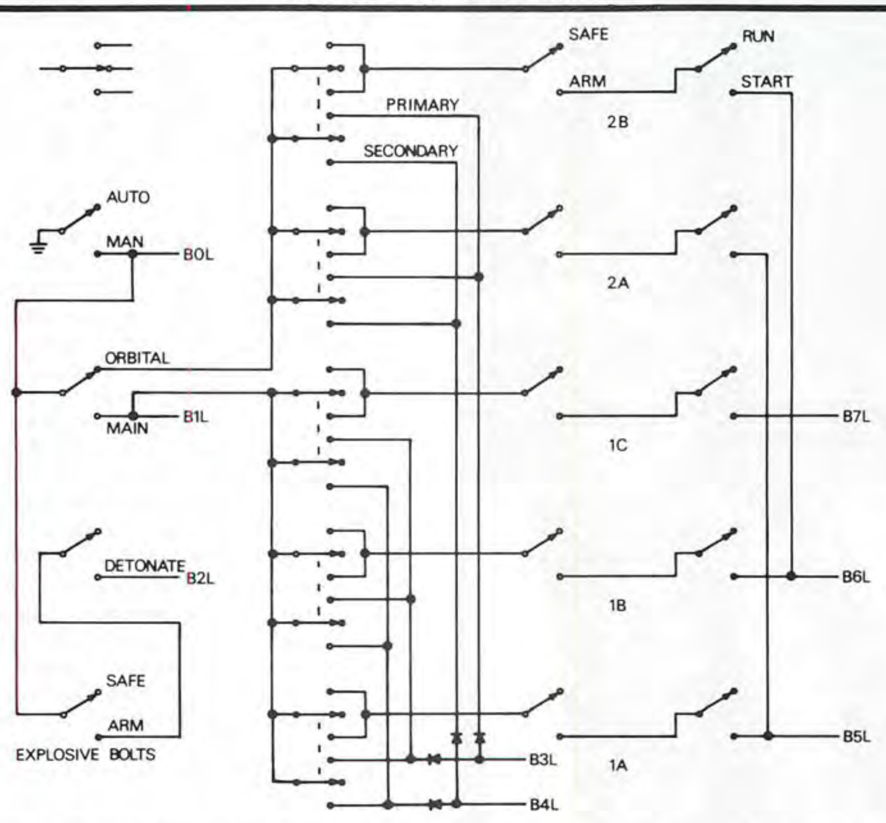


Figure 1. "Pinball Machine Logic" switch wiring (front panel seen in Photo 5).

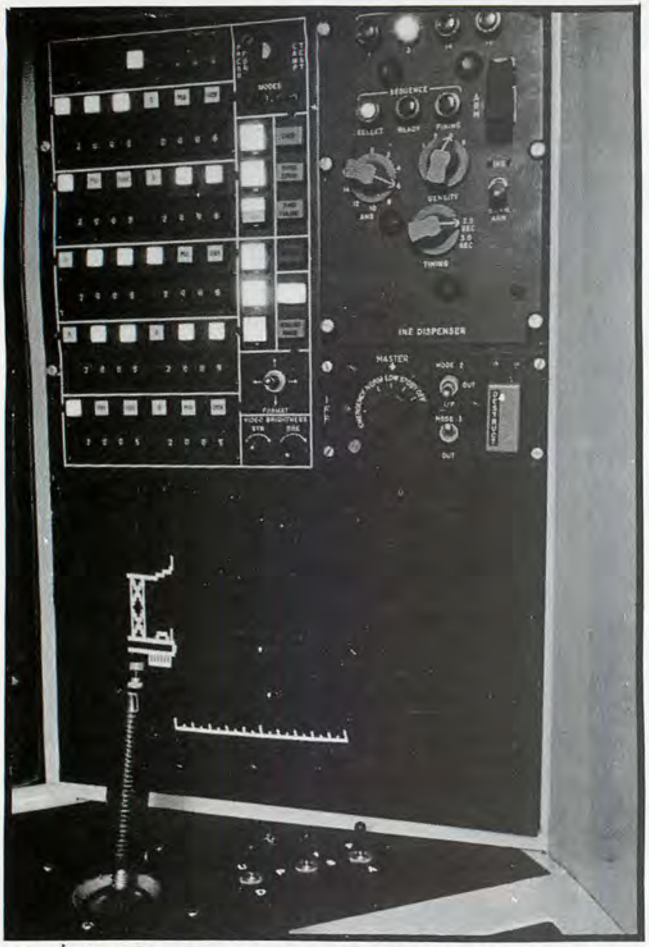


PHOTO 4 Some starboard side controls and displays.

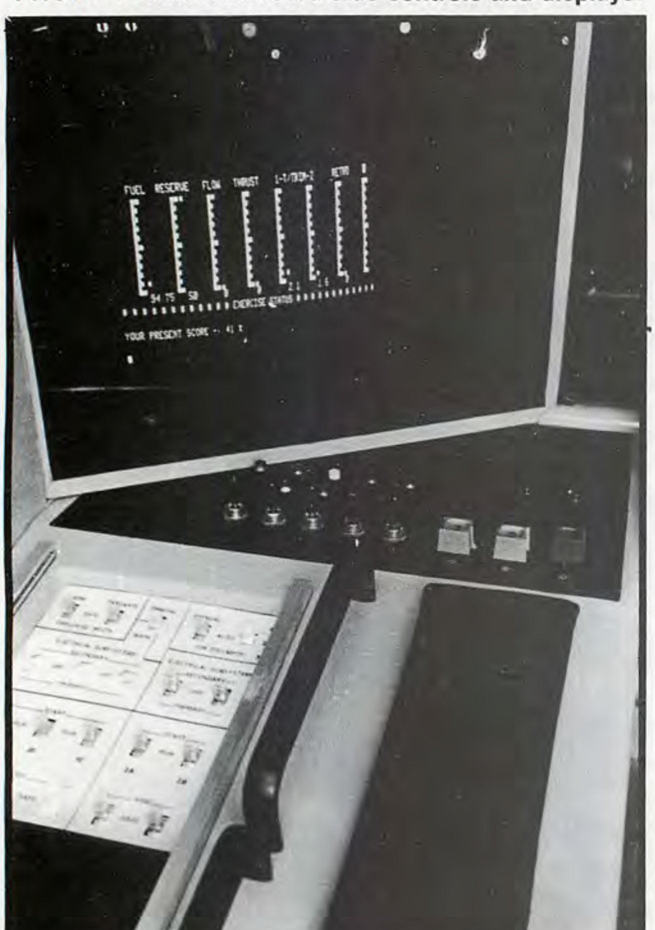


PHOTO 5 Some port side controls and displays.

we already owned. Although we now had the restriction of keeping the "officework" *capability of the processor,* we feel that the knowledge we absorbed while adapting it more than made up for this handicap.

Our final requirement was that the unit be **portable**. After all, if this project turned out half as well as we foresaw, it would be something we would want to show off. We would want to take it to hobbyist meetings, conventions, possibly trade fairs. We decided to make the ship an assemblage of individual boxes, each with only a few interconnecting wire runs and an AC input. For example, in the port console, the switch illumination power transformer is internal. It also includes the indicator lamp driver card. The switches and the analog I/O card each connect to the processor with a 16-wire ribbon cable; the overhead speaker outputs (port forward and aft) and a low voltage power cable each have individual connections.

Our past experience in providing sound recording services for live concerts had taught us that time was an important factor when setting up anything, so we decided that the entire simulator, including the rolling chair, should bolt together with identical (¼-20 x 2 in.) bolts and a few assorted screws, making the only tools needed for assembly to be a 7/16 in. nut driver and the ever present Swiss Army (27 tools in one) knife. The script was complete; now the real work would begin.

ASSEMBLY BEGINS

The first serious work on the project began with the New Year. We began to gather the major surplus components and "trial fitted" them to a design we had not yet finalized. In March we drew up the plans and purchased the raw lumber necessary to the project. It was then that we learned that it pays to have a friend with a woodworking shop; the project would have stopped dead for lack of a table saw and the expertise to use it. Although we tried to engineer the pieces so that they could be cut with a minimum of waste, some odd angles crept in. The sloping consoles, instrument clusters and rolling chair all had to be "human engineered." We found that cutting these odd angles was our biggest problem.

We discovered that freeing the main CPU from drudge work and allowing it to concentrate on the simulation itself was both a trial and a blessing. While it did make some special hardware and software necessary, it gave us a clear goal and prevented some "wandering engineering" which might have otherwise occurred. We used mostly inexpensive common parts, buying only those specialized pieces which were multifunctional.

The simulation program for the "SFS Walletsize" is written in PolyMorphic Systems BASIC Version A00. Within this program are calls to machine language subroutines, and *peek* and *poke* statements to control the interrupt structures and I/O devices which make a large number of I/O operations possible.

Peripheral chips on the bus include one UART I/O (General Instruments), one USART tape I/O (Intel), four ACIA terminal and printer I/O (Motorola), four CTC, sixteen programmable counter-timers (Zilog), one PIA 16-bits parallel I/O with handshake (Motorola), and sixteen 8212 128 bits parallel I/O (Intel). The system also contains three PolyMorphic video boards.

40K of memory is available at present, all on 4K cards from various manufacturers. Some I/O cards were purchased and heavily modified; others were built from scratch using a Vector wiring pencil. Timekeeping within the simulation is provided by a crystal-controlled hardware time-of-day clock, as well as the real-time clock function of the PolyMorphic Systems BASIC.

Analog and audio output functions are performed by the Zilog CTC chips and by interface circuits of our own design. Switches and lights are interfaced through parallel ports and more of our own circuits. All cards are

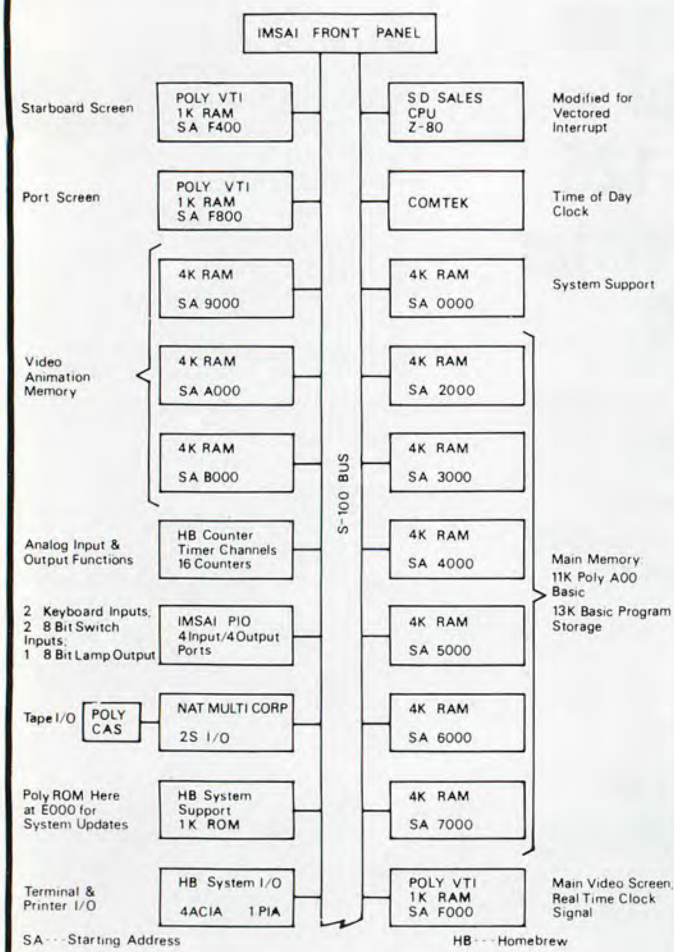


Figure 2. System block diagram.

connected to the S-100 bus in a modified IMSAI chassis. The CPU is a modified *Z-80 card (S.D. Sales), running at 2* MHz (see Figure 2).

We played so many software and hardware tricks with the computer that we finally had to erect a sign which told the public, "It's not all done with mirrors," and let me add that we couldn't have done it without PolyMorphic Systems products. The combined effect of their BASIC operating system and their video cards was unbeatable. They made it simple to control output ports, and easy to read input ports. Interfacing to machine language commands was relatively simple, and it was essential to have their interrupt and real-time operation. The BASIC graphics statements and the graphics video cards worked well together and enabled us to produce a reasonably realistic three-dimensional simulation in record time.

SIMULATION SOFTWARE

The simulation programs were written three days before the public maiden voyage of the "SFS Walletsize." These programs, running at PC '77, were a 3-dimensional Lunar Lander program, and a docking simulation. We deliberately simplified the Lunar Lander program so that passers-by could fly the ship; this included two axes on the joystick (forward/aft, port/starboard) and one slide potentiometer (descent engine variable thrust). Three video screens were used to show the lander and landing platform, displaying port/starboard motion on the main screen, with forward/aft on the port screen and an overhead view on the starboard screen. Six small speakers provided audible feedback for the rocket motor operation; all other switches and lights were inactive. The average simulation runs about two minutes; out of 200 or so people who flew the ship at PC '77, only two landed successfully.

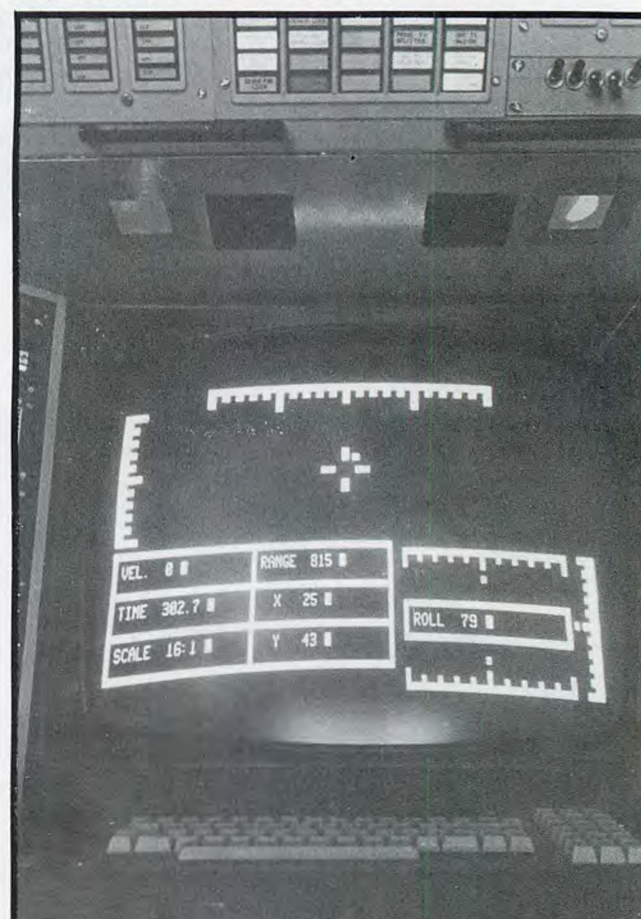
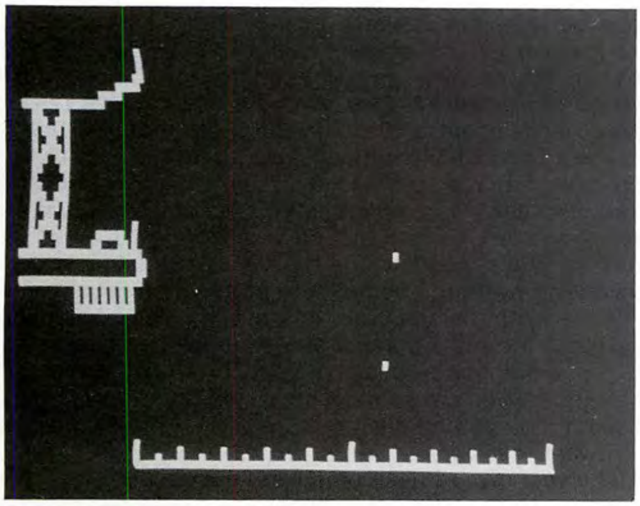
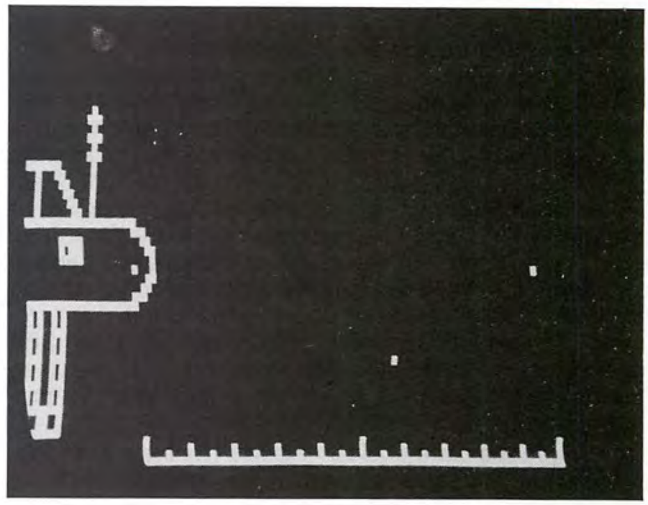


PHOTO 6 Main screen showing docking simulation.

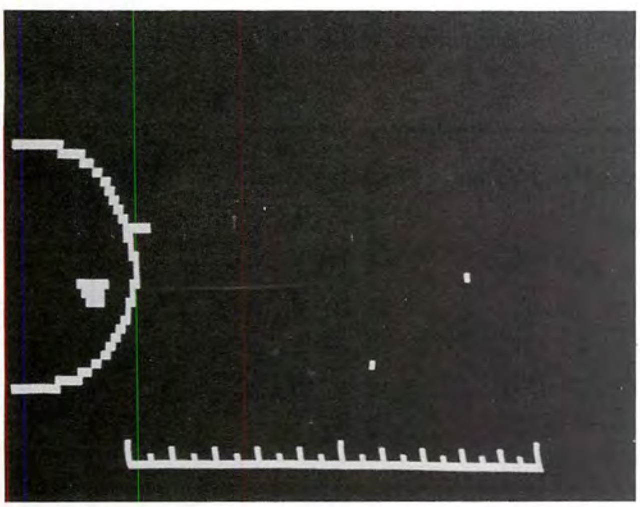


7a

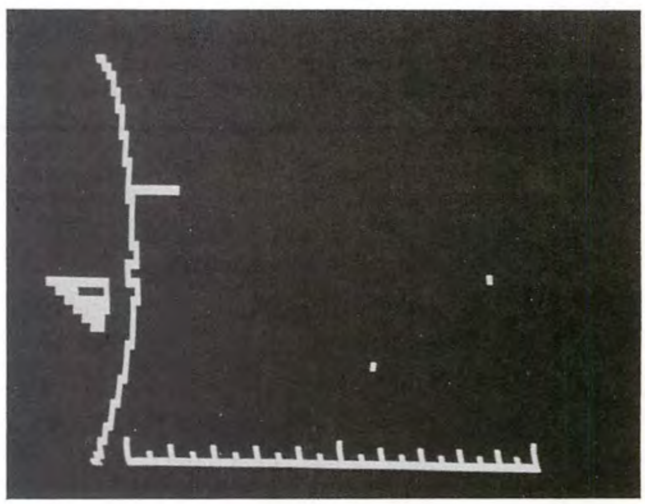


7b

PHOTOS 7a-7d Animation series showing side view of simulator (small square) moving progressively closer to command ship docking module (note the Phantom Glitch, the square near the bottom of the screen, present during the photo session and not seen since).



7c



7d

The docking simulation averages about six minutes, and was too complex to teach to exhibit-goers, so we demonstrated this one ourselves. This program uses all three axes of the joystick, one slide pot and a button on the joystick for variable retro-thrust, two slide pots and one switch for the main engine thrust and ignition. The main screen (see Photo 6) showed a number of instruments (elapsed time indicator, range, X and Y displacement, roll angle, forward and three directional velocities) as well as an alternating display of cross-hairs and boxes, incrementally sized according to distance and a scale factor. The port screen showed more instruments: main fuel, reserve fuel, fuel flow, total thrust, main engine thrust settings and retro thrust. The lower half of the screen showed the exercise score (based on time, accuracy and fuel conservation) and displayed general computer initialized comments. The starboard screen showed a side view of the docking module in four progressively larger and more detailed views, with the "SFS Walletsize" indicated by a plotted dot on the screen (see Photos 7a-7d). Pairs of lights were tied into the processor and blinked in random sequence.

Although the planned software kept us busy until "departure time," we did have a few bright ideas while we worked; these were temporarily shelved but could be added at any time. One of these encompassed the possibility of using videotaped scenes from the landing of the actual space shuttle, combined with 300 baud information on the audio track. The result would be an "autopilot" landing with a "live" outside view. The "SFS Walletsize" does not have to be a spaceship; it could become a Cessna, a Piper or even a blimp! And how about a submarine? With BASIC running in the processor, all we need is time.

APPRAISAL AND CONCLUSION

Thanks to our previous estimating experience and careful scrounging, the project came in well under budget. We'd like once again to thank our surplus dealers for many super buys. Of course, the planned-for simplicity of design and careful engineering were a major factor in achieving this cost underrun, (see Tables II and III). But — we did have a major time overrun. The lure and fascination of the project caused us to put at least 100% more time into it than we had expected. The Walletsize continued to be, to paraphrase a boater's phrase, "a large hole lined with wood into which one throws handfuls of time." We are continuing to write new programs and improve old hardware. Fortunately, many of the program modules and subroutines are interchangeable between simulations.

However — the amount of fun we've had, and the knowledge we have gained building this thing far outweigh any disadvantages. We had long felt that microprocessors could and should do more than run just a keyboard and CRT. Eventually, we would like to have the processor running a zoned (by room) program of fire and intrusion detection, environmental control, etc., with a telephone link to local police and fire departments, all running transparent to the high level language or disc operating system. The building of special I/O hardware and the writing of the interrupt driven machine language software for this program have given us the opportunity to test a number of ideas we had. The simulator project has thus given us two valuable spin-offs: a large data base of component/design reliability information and a software test bed.

Our real reward has come from the people who have seen, and flown, the "SFS Walletsize." The open mouths and staring eyes remind us of kids at Christmas. We were delighted by the crowds at PC '77 and talked ourselves hoarse. They, in turn, gave us suggestions for applications of the entire simulator, as well as specialized uses for selected portions of the software and hardware.

AFTERTHOUGHTS

We've already begun to consider making some improvements on the "Walletsize," and indeed consider building a second. We would like to add a real pilot's seat, should we find one. We are going to add a tonneau cover to both the front and the back, so that it can be protected from the dust. We wish we'd had more than 48 hours to put it together the first time, but our shooting schedule was a tight one. On the whole, we're moderately satisfied. Now that we have given you the general outline of what the simulator is and does, we are now going to cover the complexities of building the machine.

Branched to Page 108

	Time (Hrs)	On Hand Value	Purchase Cost
Wood	100		$ 78.96
Hardware (mechanical)		$ 7.00	$ 11.21
Paint & Spackle	80		$ 29.09
Finishing	26	$ 4.50	$ 23.95
Line Wiring	8	$ 7.00	$ 31.10
Controls and Displays	50	$ 46.40	$233.75
Special Hardware (elec.)		$169.00	
Aircraft Surplus	40	$ 14.00	$ 27.00
Design and Acquisition	60	$ 5.00	
TOTALS	364	$252.90	$435.06

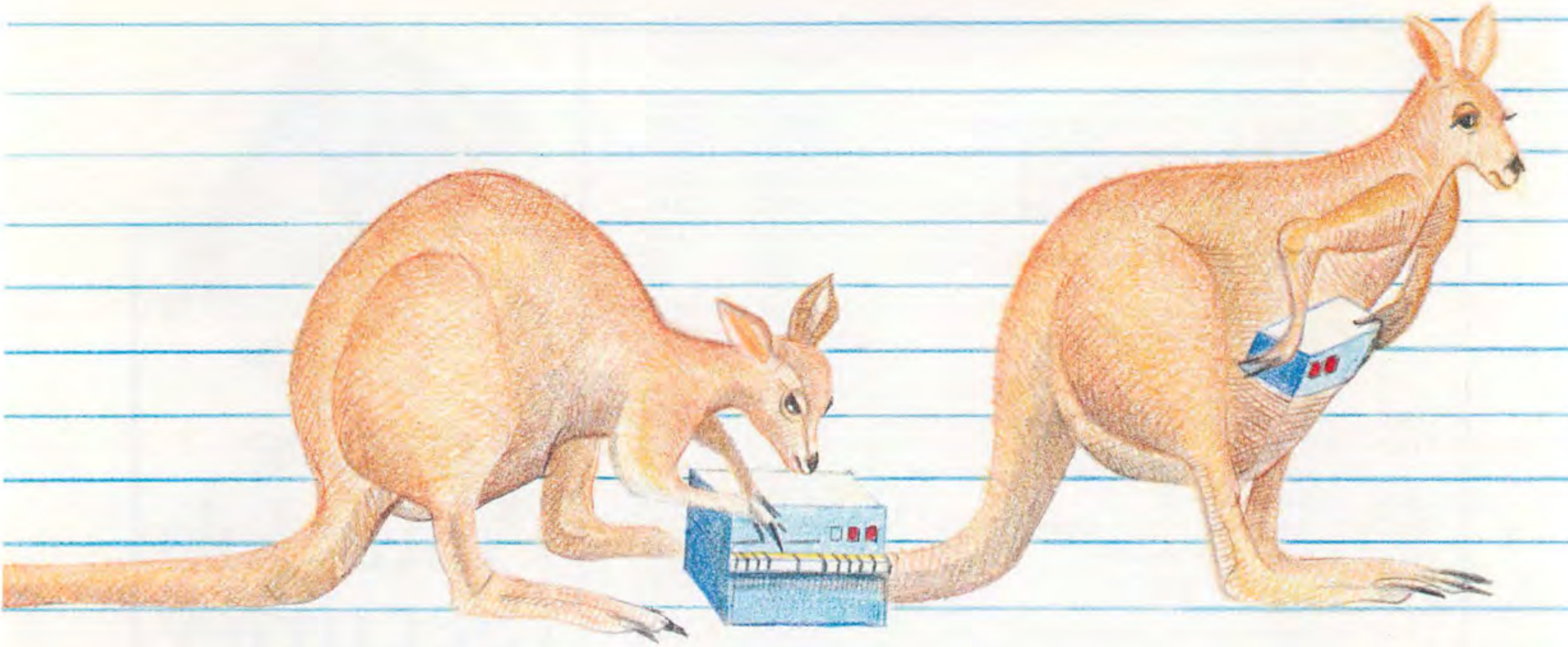
Table 2. Time and Cost Breakdown

Qty	Description	Each	Ext
6	2" x 4" x 8' lumber	$ 1.76	$ 10.56
4	4' x 8' x ½" mahogany plywood	15.50	62.00
1	4' x 8' x ¼" tempered masonite	6.40	6.40
	misc. hardwood stringers	on hand	5.00*
64	¼-20 x 2" machine screws	.042	2.69
64	T29 x 450 tee nuts	.03	1.92
8	6-32 x ½" machine screws (s/s)	.054	.44
8	6-32 tee nuts	.02	.16
	misc. nails	on hand	2.00*
	misc wood screws	on hand	6.00*
	white glue	on hand	1.29*
	spackle	on hand	1.85*
1 gal.	gray polyurethane paint	12.95	12.95
1 qt.	flat black paint	3.85	3.85
	misc. expendable painting supplies	9.15	9.15
2 yds.	black canvas	3.00	6.00
1200 cu"	foam padding (for chair)	2.00	2.00
345 sq"	smoked plexiglas	5.20	5.20
1 pr	appliance rollers	3.25	3.25
8	black metal handles	.50	4.00
2	pieces, alum. extrusions	1.00	2.00
	misc. cable harnesses & fittings	on hand	3.00*
	vinyl stick-on letters	on hand	1.50*
2 yds.	black adhesive-back vinyl	.75	1.50
3	duplex boxes with outlets	5.11	15.33
1	15 amp extension cord	8.00	8.00
3	2-out to 6-out convert-a-plug	.79	2.37
4 ft	braided shield (ground plane)	.10	.40
4 ft	Romex cable	on hand	1.00*
6	line cords	on hand	6.00*
4	cleats & 7½ watt 110V AC bulbs	1.25	5.00
1	5V DC power supply	24.95	24.95
1	18V AC 3 output transformer	on hand	10.00*
2	6V AC transformers	on hand	6.00*
1	24V AC transformer	on hand	8.00*
6	0.3 watt speakers	1.80	10.80
50	push-push illuminated switches	.10	5.00
34	bat handle toggle switches	on hand	3.40*
25	illuminated toggle switches	on hand	5.00*
1	3 axis joystick w/mods	6.00	6.00
3	linear attenuators	on hand	9.00*
10	24V lamp assemblies	on hand	5.00*
1	keyboard	25.00	25.00
1	25" diagonal monitor (incl. repairs)	47.00	47.00
2	9" diagonal monitors	40.00	80.00
1	Gemini mode display	5.00	5.00
2	Apollo mode displays	15.00	30.00
2	analog interface cards	8.00	16.00
3	lamp driver cards	6.00	18.00
	misc. cables & connectors	on hand	50.00*
1	S-100 bus interface card ($60 LSI, $6, $19 card)	on hand	85.00*
2	radio compass controllers	2.00	4.00
1	3 digit readout and switch	on hand	4.00*
1	ATC transponder	5.00	5.00
1	INT control box	4.00	4.00
1	ATC controller	10.00	10.00
1	INE dispenser	3.00	3.00
1	IFF control box	1.00	1.00
	misc. indicators & displays	on hand	10.00*
	TOTALS		
	On Hand	$247.90	
	Out of Pocket	$435.06	

*approximate cost if purchased

Table 3. Materials

CIRCLE INQUIRY NO. 54



Social Impact of the Computer

"It has often been pointed out that when something changes by a factor of ten (often referred to as an order of magnitude), it has profound effects on society. For example, a person can walk about 4 miles per hour, travel by automobile at something like 40 miles per hour, or go by jet aircraft at roughly 400 mph. The Apollo astronauts averaged about 4000 mph in their trips to the moon. Each of these differs by a factor of 10 — 4-40-400-4000 — from walking to space travel in three orders of magnitude. The increase made possible the flight to the suburbs and drastically changed our mores. The jet aircraft has shrunk our world immeasurably. The impact of space travel is just beginning to be felt.

"In the computer field even greater orders of magnitude changes are taking place in a short period of time . . . by 1984 the cost of computing power will decline by at least another factor of 100."

(Wessel, *Freedom's Edge,* p. viii)

Imagine what would happen if someone predicted that a familiar item would decrease in cost by a factor of 100 by 1984. A family car for \$60, or a house for \$600! The computer itself can be regarded as a means of increasing human computational power. It can do in a ten-thousandth of a second a computation that would take a man several seconds. This hundred thousand-fold increase in speed is five orders of magnitude. It is almost impossible to obtain accurate figures, but to a first approximation, it can be said that in the thirty years between 1950 and 1980:

- The cost of hardware—the computer itself—has dropped about three orders of magnitude.
- The speed of computers has increased by about three orders of magnitude.
- The cost of computing has dropped by about five orders of magnitude.
- The power of computers has increased by at least three orders of magnitude.
- The computational speed of computers compared with the human brain is about six orders of magnitude greater.

The above figures are intended as a general guide. Because of the tremendous development of computer technology, accurate figures are impossible to obtain. I would appreciate knowing about any authoritative statistics that are available.

If three orders of magnitude change in the speed of transport have had a profound effect on our society, how can the change implied by the above figures be described? There comes a time when the amount of change becomes so large that there is a quantum jump, a totally new effect. I would suggest that the changes caused by the introduction of the computer will amount to a quantum change in our society. The term "revolu-

By Timothy Mowchanuk

An Editorial from "Down Under"

tion" is almost too quiet, too small a term to describe the effect. The computer has become so ubiquitous, has so pervaded our society that, paradoxically, the computer revolution has gone almost unnoticed. Just what the total impact of the computer will be, what changes it will make to our society are almost impossible to predict. The revolution is still too new for accurate prediction. In my opinion, it will have more impact and a greater effect on society than the advent of the automobile and television combined. When the automobile was first introduced who could have predicted the changes it would make to our society?

Computers are pervading our society and personal lives to an astonishing extent. This is especially true of the microcomputer, the computer-on-a-chip. Virtually any mechanical or rote operation is susceptible to computerization. When thinking of computers, it is normal to think of big machines. The concept of the computer is popularized in science fiction. It is the big uses that come to mind; the computer in a bank, the computer in administration printing out payrolls, the computer in government. But computers have been getting smaller and smaller, and cheaper and cheaper until we now have the powerful microcomputer. It is now cheaper to have a microcomputer control a washing machine than to use traditional mechanical control. Computer-controlled washing machines, sewing machines and kitchen stoves are now being produced. The author actually used a computer-controlled oven while in the United States last year. It should not be more than a year before this type of item will be available in Australia.

Computerized television games, much more sophisticated than the standard "TV Table Tennis," are already a reality. There are several available in the United States for slightly more than $100. Such dedicated microcomputers are not the only form of personal interaction with computers. The general purpose computer is almost a reality. A general purpose home computer is on sale in the United States for about the cost of a colour television set. It is intended to sell this machine as a home appliance. It is sold in home appliance stores exactly the way television sets are sold — buy it, take it home, plug it in, use it.

As teachers and educators, we cannot afford to ignore the computer. A student of today who is not familiar with the computer is at a grave disadvantage in trying to cope with his personal and business worlds. This is not to say that I advocate teaching computer programming to all students, nor that computer programming should become a major part of the high school syllabus. I believe this to be a "non-issue" that will automatically answer itself with time. I am strongly suggesting, however, that all students must have some sort of "computer literacy." The effective person in the future society must be familiar with computers. It is vital that he have some appreciation of computers. He must overcome the fear, suspicion and the mystique that presently surrounds computers and their use. He must appreciate what computers *can* and *cannot* do, their inherent limitations and the problems associated with their use — privacy, data banks, computer crime, inability to make value judgements. Computers will cause a number of social problems and the citizen of tomorrow of every nation must have at least a basic knowledge of them to understand and cope with these problems.

LSI-11 Microcomputers

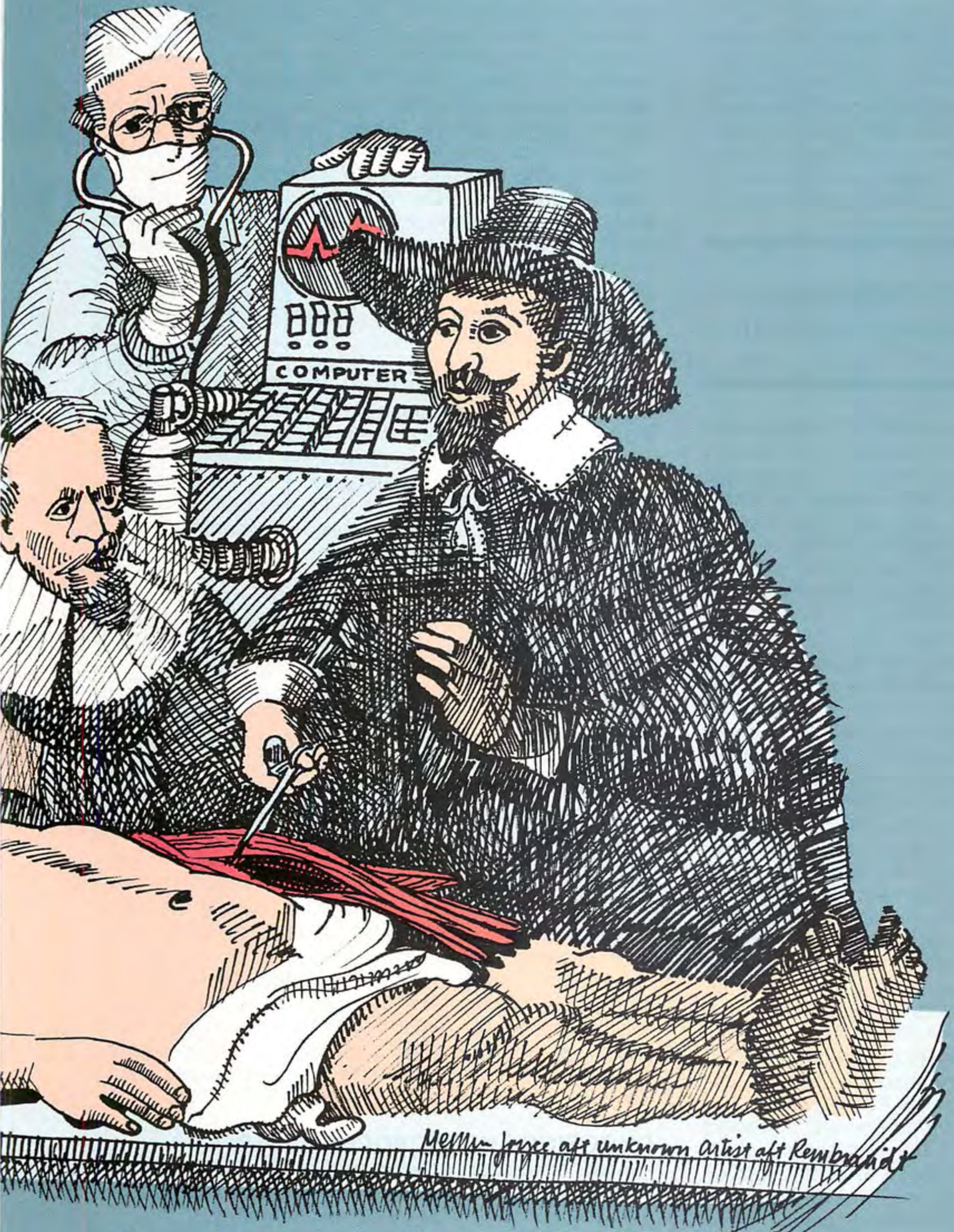
BACKGROUND

One of the most dramatic examples of medical science at work is the intensive care unit (ICUs) at a modern hospital. Based on the connection of patients to the latest electronic equipment, including an increasing number of computers, ICUs often mean the difference between life and death to the critically-ill.

Originally developed from the post-operative recovery room, where the patient was observed during the period immediately following an operation, intensive care units provide a system whereby the continuous monitoring of acutely ill patients enables a swift response to adverse changes in the patient's condition during the three to five days required by the body to stabilize.

Every key factor affecting recovery is under constant scrutiny and an overall watch is kept by a computer. Programmed to evaluate and compare every reading fed into it by the monitoring instruments, the system flashes a warning to medical personnel whenever measurements show undesirable variations. A well-introduced and accepted computer system handling constant monitoring allows any change instantly to be dealt with. This is particularly important in the case of cardiac arrest. Unless the heart is restarted in three to four minutes the patient's brain will be starved of oxygen causing death or permanent brain damage.

The costs of such complex systems are naturally high, but can be recovered. The removal of critically-ill patients



in Hospital ICU's*

By Ed Heesbeen

from general wards to ICUs enables the rest of the hospital to function more efficiently, while the concentration of patients with the most potentially lethal conditions permits an increased number of lives to be saved.

Out of every 100 patients admitted to general wards after a myocardial infarction, about 30 die. In an ICU this toll is halved. In a general ward only about one patient in ten survives serious chest and lung injuries; in an intensive care unit six out of ten are likely to survive.

ROTTERDAM THORAX CENTRE

The ICU at the Thorax Centre of the Erasmus University Academic Hospital in Rotterdam, Holland, is an excellent example of computerised monitoring. The Thorax Centre has at present an 18-bed Intensive Care Unit divided into three parts — post-operative care, coronary care and respiratory care. Since a single patient in just the post-operative care area can give rise to more than a dozen signals that need virtually continuous monitoring, it is obviously impossible for the nursing staff to handle this workload and still perform other nursing chores. In fact, a recent work analysis showed that more than half of the time was spent looking at equipment and attending machines while only 11% was left for actual nursing. The large size of the Centre's ICU makes the use of a computer necessary.

*Reprinted from Microcomputer Data Publications, May 1977, Amsterdam, Netherlands.

For the past five years the Centre has made use of a highly specialized system running on a DEC PDP-9 computer using a "stripped" PDP-15 processor to handle electrocardiogram (ECG) pre-processing. The PDP-9 is now being used almost to the limit of its CPU capacity and it is difficult to add any new applications because of lack of core-memory and CPU time. This, added to pressure from the clinical staff, who would like a wider range of applications than currently provided, has led to the Thorax Centre's decision to replace the system with a network containing LSI-11 microcomputers.

...instant reference included a patient overview showing heart rate...

NETWORK SYSTEM

The heart of the network will be a PDP-11/10 minicomputer equipped with 16K words of memory, a 1.2-million word cartridge disc, a console terminal, a real-time clock, a number of interactive video-display terminals and a serial line multiplexer interfacing to four remote LSI-11 microcomputers located in the intensive care units. Beside each bed in the ICU areas will be a further eighteen LSI-11s connected via a DVL-11 interface to the appropriate nursing station processor.

The bedside LSI-11s each connect via a DRV-11C 16-bit parallel input/output interface to a number of general input modules and a display module all of which are contained in a neat 19" x 24" unit known as a Uni-box.

Each of the five input modules in the Uni-box is multi-purpose and contains an A/D converter and a programmable 32-character touch-sensitive LED-display. Individually, the five modules can handle ECG, arterial pressure, blood temperature, right atrial or left atrial pressure, fluid balance, cardiac output and respiration according to needs. The separate display module can display, under computer control, up to four independent signals either as a moving trace or as a frozen image.

The bedside LSI-11 provides all continuous monitoring functions required for a given patient, displays the signal traces and parameter values and generates an audio alarm should the parameters be exceeded. It also allows the nurse to change the threshold values for alarms and to calibrate transducers by touching the appropriate function on the LED-display. Parameters are reported every minute to the central PDP-11/10.

The nursing station microcomputers are equipped with a number of display modules and a strip chart recorder interface enabling the nurse to select soft- or hard-copy output of any patient signal. In addition, each nursing station has an interactive video-display terminal which is directly connected to the central PDP-11/10. This terminal can be used to obtain patient status information, census information, graphical displays and other terminal-oriented procedures.

The central PDP-11/10 is responsible for maintaining a data-base for each patient, generating patient reports and providing program service to terminal users. The system is designed in such a way that the bedside computers can work independently of both the nursing station and the central computer. This increases overall reliability and ensures that failure of any component in the system can only degrade performance but not bring the whole system to a halt.

SOFTWARE

Software for the network system and other systems functioning within the Thorax Centre is GNOME-11, General-purpose Network Oriented Multi-user Executive, developed by the Centre's own Department of Experimental and Clinical Information Processing. It runs on any PDP-11 computer with between 8K and 28K words of memory, serial interface and a clock. Supporting MACRO-11 and FORTRAN-IV based applications, GNOME represents several years of work-hour effort on the various systems developed by the computer department since it moved into the Thorax Centre in 1968.

"During the middle 1960's ICUs became generally established in Europe," says Professor P.G. Hugenholtz, head of the Thorax Centre, "and we decided straightaway that our philosophy should be the dedicated approach of using one computer system per application. We tailored software to our own needs and built the system accordingly. By the time the Thorax Centre was opened for patients in 1971, we already had a working monitoring system."

The working system to which Prof. Hugenholtz refers was the original 32K PDP-9 with 750,000-word fixed head disc, special communication interface between the PDP-9 and PDP-15, 32-channel video-display system, 64-channel A/D converter, 32-terminal multiplexer, five DECtape units and two teletypes. This installation enabled each bedside to have a TV monitor and keypad and each nursing station to have a TV monitor, keypad and bed-selector pad. Information available for instant reference included a patient overview showing heart rate, arterial pressure, temperature, etc. refreshed every 20 seconds; patient condition giving more detailed information about ECG, hemadynamics and temperature. Also included were graphics of thirteen different conditions either in time trend form or histogram form and patient status showing how long the patient had been on the computer system as well as which monitoring he was undergoing.

In 1972, a PDP-11/20 was installed for on-line catheterization preceding open-heart surgery. It enabled cardiologists to obtain results immediately rather than hours later. Since then the value of computerisation has been proved by the successful conclusion of 1500 catheterizations with reduced manpower and increased accuracy.

Subsequently, a PDP-11/40 MUMPS (hospital utility multi-purpose programming system) was installed to handle work-listing, patient scheduling, administration and to provide a common data base for all the other systems. Additionally, experimental work is nearing completion on a PDP-11E10 system with A/D converter, video-display and DECwriter for analysing electrocardiograms during exercise.

CHOOSING THE MICROCOMPUTER

Apart from enabling all the monitoring equipment to be included in one neat, easy-to-control bedside Uni-box unit, the LSI-11 offered other advantages.

It gave them software compatibility with all the other PDP-11s they are using. This was extremely important because in an application of this nature software costs are often more than hardware. The initial low cost of the installation enabled the hospital to put one beside each bed which could not have been affordable with minicomputers. More CPU capacity per patient is obtained with greater reliability because of its stand-alone capabilities. Programming time was also saved because using distributed processing enables them to simplify data handling for the individual applications. Furthermore there is still space to add all the application programs that are needed and yet still leave enough room to cope with any future demands.

The network project for the intensive care unit was started in early 1975. By the end of this year the first LSI-11 Uni-box will be up and running in the post-operative area. Full system implementation is planned for 1977.

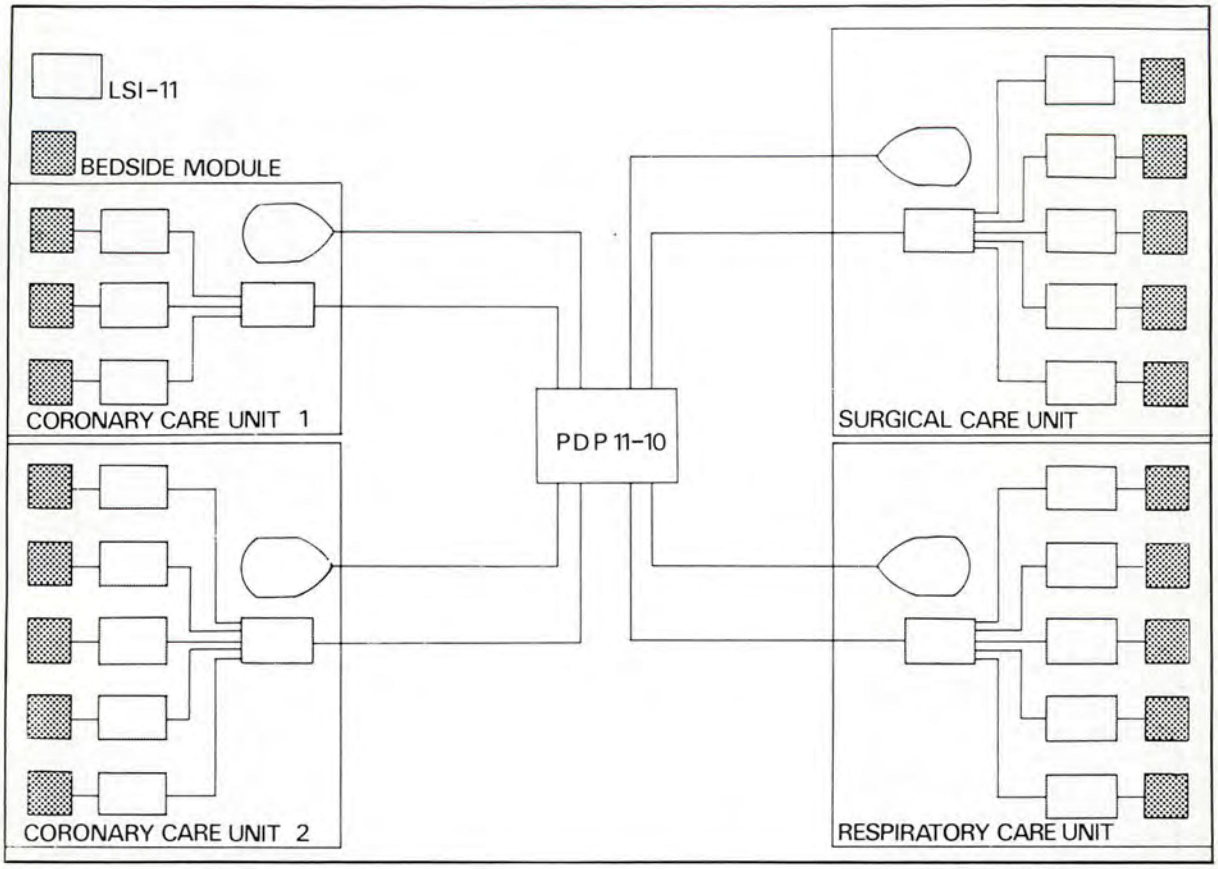


Figure 1. Diagrammatic outline showing the simple tree structure of the LSI-11/PDP-11/10 network at the Thorax Centre.

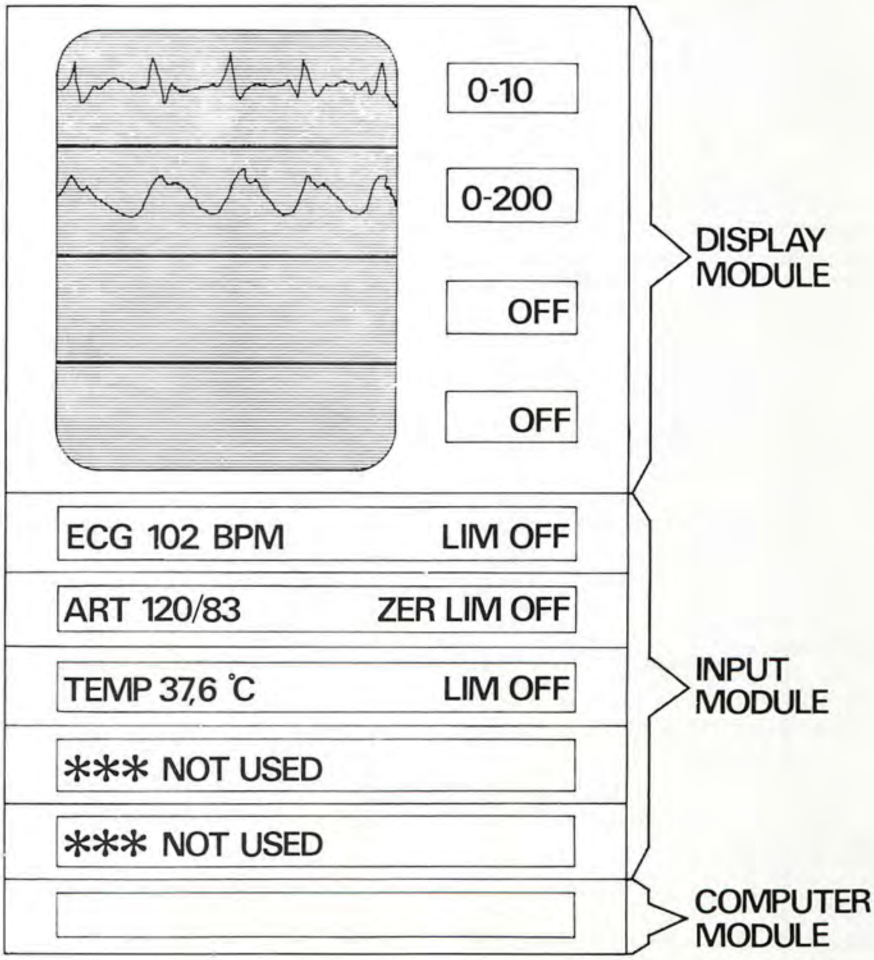
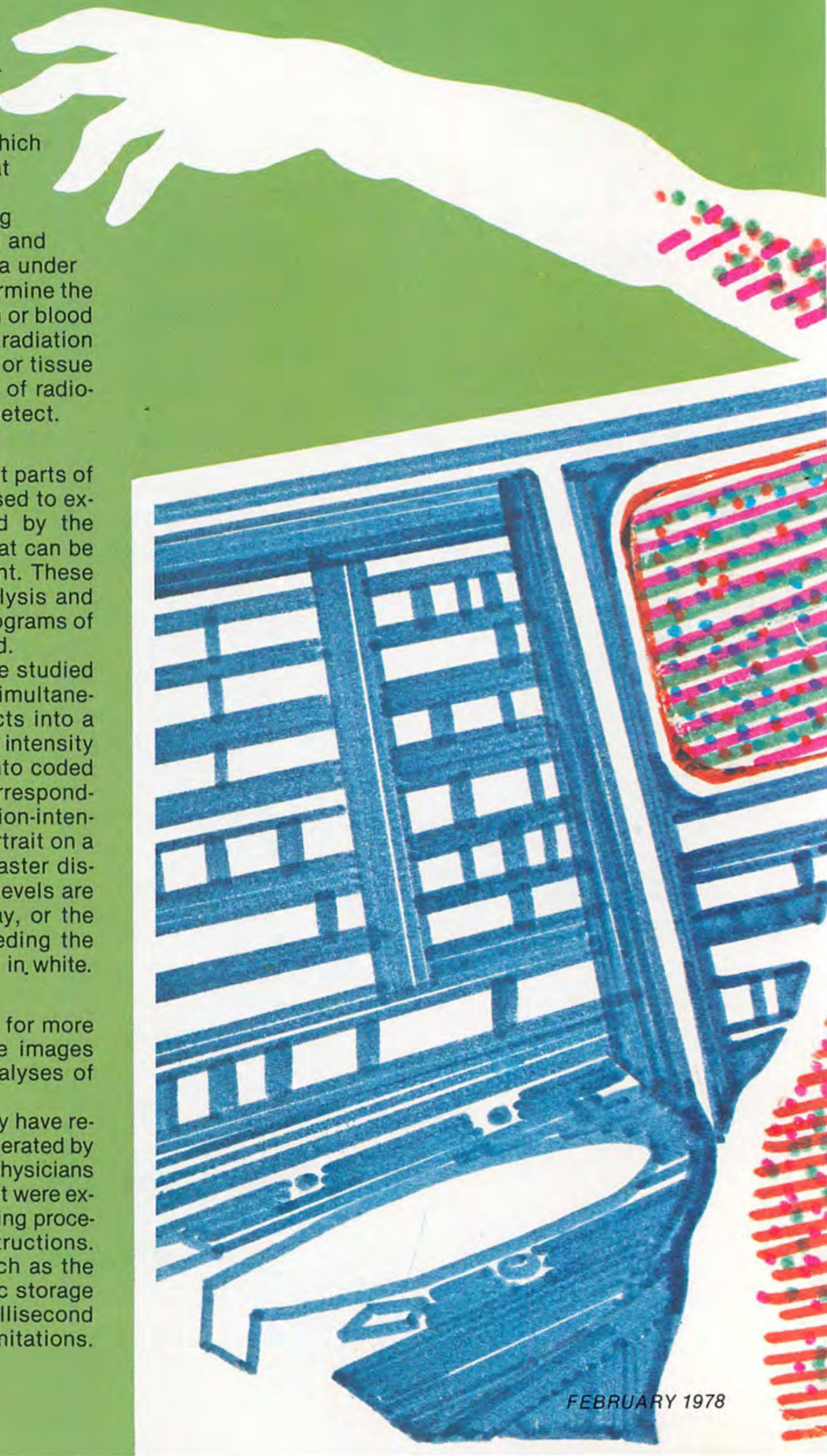


Figure 2. The bedside Uni-box contains an LSI-11 microcomputer, five multi-purpose programmable 32-character touch-sensitive LED display input modules and a display module.

Pertec Disk Diagnosis in an



INTRODUCTION

Diagnosis by means of a gamma camera provides a painless technique for studying the various organs and blood vessels of the body. The Elscint CDT-2 camera is a TV-like device which is sensitive to gamma radiation, especially that emitted by radioactive isotopes.

The diagnostic technique involves injecting the patient with a harmless radioactive drug and when the radioactivity has perfused the area under observation, the camera scans the area to determine the amount of radiation present. The healthy organ or blood vessel readily permits the drug to circulate and radiation levels are low. Abnormalities such as growths or tissue damage absorb the drug leaving higher levels of radioactivity which the gamma camera can easily detect.

THE PORTRAIT

Different drugs are used to examine different parts of the body. For example, radioactive iodine is used to examine the kidneys. Each kidney is scanned by the camera which produces an electrical signal that can be used to produce a portrait of radiation present. These pictures, called 'renograms', are used for analysis and comparison, either with each other or with renograms of other kidneys known to be healthy or diseased.

The camera scans the area of the body to be studied in much the same way as a TV camera and simultaneously converts any radiation particles it detects into a signal whose amplitude varies according to the intensity of radiation present. This signal is digitised into coded blocks of pulses, each block representing a corresponding radiation particle count (i.e. level of radiation-intensity). These pulses are used to produce the portrait on a CRT. The portrait is in the form of a TV-like raster display in which the different radiation-intensity levels are represented either by different shades of gray, or the eight colors of the rainbow. Intensities exceeding the maximum color/shade coded level are displayed in white.

COMPUTER-AIDED EVALUATION

Although gamma cameras have been in use for more than ten years, techniques for evaluating the images they produce have been largely limited to analyses of single (i.e. static) images.

Dynamic studies have been possible but they have required complex equipment that has had to be operated by physicists or computer programmers, because physicians lacked the know how. The studies they carried out were extremely time-consuming because of slow operating procedures involving keyboard entry of computer instructions.

The advent of low-cost microcomputers such as the Elscint CDP-2 and high capacity magnetic disc storage such as the Pertec D3000 with its 30-40 millisecond average access times, has overcome these limitations.

Speeds Up Israeli Hospital

By Linda Folkard-Stengel, Feature Editor

Now the system can process and store images at ten frames a second thereby enabling them to be displayed within a fraction of a second after the camera has produced them. The physician can then observe the passage of the drug through an organ such as heart and lung and map its progress through the right atrium and ventricle, the pulmonary artery, each lung, the left atrium and finally the left ventricle, virtually as it takes place, and can immediately assess the condition of these various parts.

The equipment can record up to 200 frames in sequence, automatically annotating them with identification data such as patient number, study data and frame number, storing the data on a disk.

PUSH-BUTTON OPERATION

From an operating point of view, the Elscint CDP-2 has eliminated the need for computing skills and can be used by physicians with the minimum of instruction. Incorporating a camera-interface console and a video display processor, the equipment is turnkey. The camera can be triggered by a pushbutton-controlled pulse-generator that provides a wide range of frame-acquisition speeds, or by physiological signals from transducers attached to the patient's body. During data acquisition, statistical accuracy and reliability are assured by a special buffer memory that effectively eliminated time-lags in the system.

Image processing facilities include two methods of ensuring that camera response is uniform to within less than 1 per cent non-uniformity. Images can be displayed in color or black and white and enhanced in several ways. A choice of radiation-count scaling for the colors or shades of gray is provided, plus a smoothing technique that enhances colors/shades by averaging the count data of each display memory cell with that of the eight cells surrounding it. Any color can be removed at the press of a button when required.

DUAL-ISOTOPE STUDIES

Dynamic studies off-line are possible. Any group of frames can be displayed in sequence at selected rates and display times in forward or reverse directions. Images obtained at low-radiation-count rates can be en-

hanced by doubling frames to form a single composite image. Up to 99 frames can be integrated. Also possible are dual-isotope studies, in which the two images are superimposed one upon the other and then subtracted one from the other to highlight a particular region.

Frames which have been processed for smoothing, frame-addition uniformity correction or dual-isotope study, are identified by lights on the CDP-2 control front panel whenever they are displayed.

The Elscint system also provides the physician with histograms of integrated radiation-counts as a function of time for any region he may choose whether it is within the area of study or outside it. Up to six rectangular regions can be selected within the area of study, regardless of their size, location or whether they overlap; and up to two regions outside the area, the latter regions being monitored by external scalers. Histograms for all eight regions are displayed simultaneously, each in a separate color.

STATIC STUDIES ALSO

Although primarily designed for dynamic studies, the CDP-2 is equally suitable for static studies. In studies of the brain, for example, the display-enchancement and smoothing facilities enable abnormalities such as lesions to be clearly seen. In addition, use of a low-count scaling so that the lesion radiation is displayed as white, enables clearly visible x-profiles of the lesion to be made so that its significance can be readily ascertained.

The Elscint equipment's ability to make dynamic studies of the brain in operation enables it to determine the precise moment of death of a mortally ill or injured person.

The traditional method of determining death has been to monitor the pulse of a person's heart and pronounce him dead when his heart stops beating. The heart, however, is operated by signals from the brain. Thus a more precise criterion for pronouncing a person dead is when his brain stops working. Adoption of the brain-death definition has become necessary with the advent of transplant surgery and life-maintaining devices such as heart pacemakers, that produce their own signals for operating the heart independently of the brain and create situations where a patient's heart continues to beat for some time after his brain has ceased to function.

Lack of precise data on cessation of cerebral activity has often resulted in the loss of organs donated for transplants — eyes, heart, liver, kidneys and the like. Disagreement as to whether the patient is dead or not has delayed the removal of the donated organs until they have become unsuitable for transplant. Many forensic questions have been raised in the past years disputing whether a patient was actually dead at the time organs required for transplants were removed from the patient's body. The fact that death is a process of the brain, rather than the heart, ceasing to function, has not generally been in dispute. The difficulty has been to devise a rapid and reliable method of detecting when the brain has in fact ceased functioning.

OVERCOMES THE PROBLEM

The Elscint equipment overcomes this problem through its ability to monitor the blood-flow through the brain, the vital factor upon which the operation of the brain depends, and to detect changes in flow almost instantly. The equipment's response is such that cessation of brain activity can be detected and indicated within a fraction of a second of occurrence. With death positively established, organs donated for transplant may be removed as soon as possible and maintained in the best possible condition ready for transplanting.

Branched to Page 175

BASIC Cross Assembler for the 8080

By Peter Reece

INTRODUCTION

A cross assembler is a high level language program which assembles low level code for a goal computer. Since the host machine is usually much larger than the goal computer, code which is too long to be assembled in the smaller machine can be easily handled by the cross assembler. The binary tape which is produced can then be loaded directly into the goal computer.

CROSS

The cross assembler reproduced on the following pages is written in BASIC for a PDP-10 computer, and will accept and assemble code for 8080-based computers. It allows the user to produce ASCII listings in octal or HEX of his source code, define variables for a symbol table, and generate a binary file of the assembled source.

The program requires two passes (i.e., it reads the source code twice), hence random access files are not required. This permits the user with a smaller secondary memory, such as sequential cassette storage, to use the system. In addition, much less storage space for variables is required by the host computer.

The first pass scans each line of source looking for an OPCODE. If one is found, the address pointer a(i) is advanced an appropriate number of bytes. If a *comment only,* or an *equate* or *origin* line is found, a(i) is set equal to a(i-1), where 'i' corresponds to the previous statement number. Pass One also creates a symbol table of all labels and equates with user defined mnemonics.

Pass Two translates OPCODES and their arguments into octal, and outputs the results in both a binary file (as address, byte one, byte two, byte three) and an ASCII form (see example in Figure 1). All translation is initially done into octal. If all attempts at translation fail, byte one is set equal to '777' (i.e. an extra bit is flagged), the error count is increased by one, and Pass Two continues. If, during a listing, a '777' is encountered in byte one, a question mark is output beside the line being listed. The accompanying flow charts indicate the proceedings during the two passes. (See Figures 2 and 3)

Use of the program is straight forward. A number of options are available:

$:	The character immediately preceding the dollar sign in a line of source will be translated into octal, with the ASCII letter 'a' being translated as '001' and 'z' as '032'.
#:	If followed by a 'd', the numeric preceding the '#' will be translated into octal from decimal; if followed by 'h', the translation will be from HEX to octal; if followed by 'o', the numeric will be assumed to be octal.
:	Any ASCII string preceding a colon will be assumed to be a label, and will be entered in the symbol table.
=:	The right hand side of an equate will be translated into octal, and set equal to the left hand side as an entry in the symbol table.
/:	A slash is followed by a comment.
*:	An asterisk precedes a numeric expression which is to be used as an origin address.
?:	Occurs beside all untranslatable lines of output.
HEX:	If a user wishes output to be listed in HEX, type this when the program types 'Options:'.
OCTAL:	(default condition) — output is listed in octal if this option is chosen.
LF:	Choosing this option produces seven line feeds per 66 lines of output, thereby producing 66 lines per 9*11-inch page.
NOLF:	Suppresses LF (default option).
LIST:	Lists assembled code beside each line of source at the end of Pass Two (default option).
NOLIST:	Suppresses 'list'.
BIN:	This option produces an output file which may be read by the goal computer. For each line of source which is not a comment or an origin or equate line, 'BIN' produces the current address, byte one, byte two, byte three, and a carriage return (default option).
NOBIN:	Suppresses 'BIN'.
SHOW:	Print the source listing; do not assemble it.
END:	End the program.
MAKE:	To create a source file, type "make" when the program starts, then type your source (one line of source per input line). Terminate "make" with the line "end".

```
options:      ?hex    list  1f
      adrs    op      b2  b3
      00000                       /
      00000                       /Here's a simple do-nothing
      00000                       /example of the program.
      00000                       /
      00006                                *7#d
      00007   31      3D          start:  lxi sp'61#d        /label line
      0000A   AF                           xra a
      0000B   47                           mov b'a    /"'" is a separator
      0000C   21      08          ok:      lxi h'load  /fwd ref
      0000F   48                           mov c'b
      00010   0E      24          write:  mvi c'44#o  /octal num
      00012   21      08                   lxi h'load
?     00015   **      48                   xra j         /note error code
      00016   0E      0A          mvi c'$j               /alpha-num conversion
      00018   D5                           push d
      00019   76                  end:     hit
                                           load = 10#o  /equate line

start = 07
ok = 0C
write = 10
end = 19
load = 08

      # ERRORS = 1
```

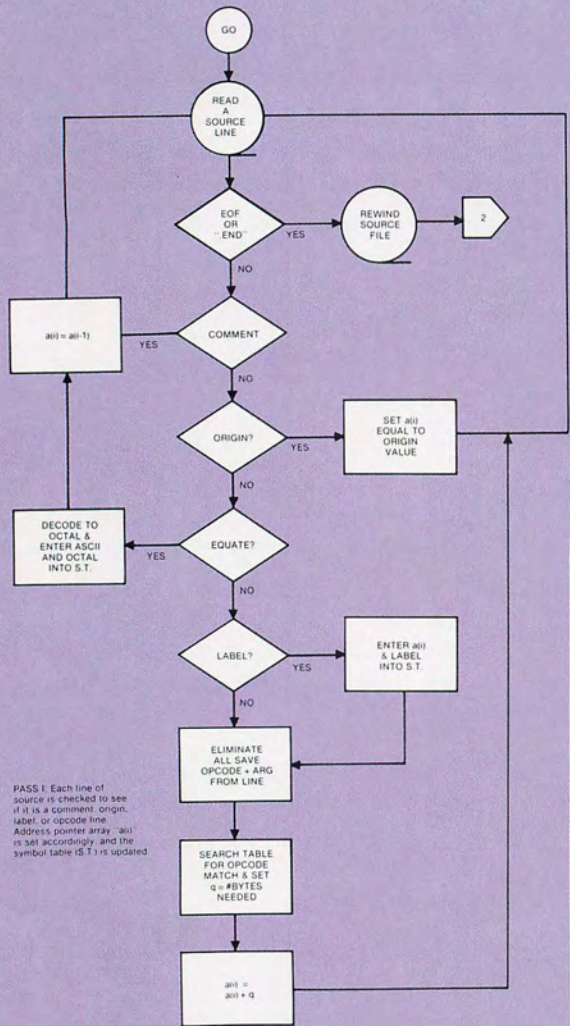
```
options:      ?octal  list  1f
      adrs    op      b2  b3
      0       0       0   0       /
      0       0       0   0       /Here's a simple do-nothing
      0       0       0   0       /example of the program.
      0       0       0   0       /
      6       0       0   0                *7#d
      7       61      75  0       start:  lxi sp'61#d        /label line
      12      257     0   0                xra a
      13      107     0   0                mov b'a    /"'" is a separator
      14      41      10  0       ok:      lxi h'load  /fwd ref
      17      110     0   0                mov c'b
      20      16      44  0       write:  mvi c'44#o  /octal num
      22      41      10  0                lxi h'load
?     25      1027    88  0                xra j         /note error code
      26      16      12  0       mvi c'$j               /alpha-num conversion
      30      325     0   0                push d
      31      166     0   0       end:     hit
                                           load = 10#o  /equate line

start = 7
ok = 14
write = 20
end = 31
load = 10

      # ERRORS = 1
```

Below is a simple example of the use of the program:

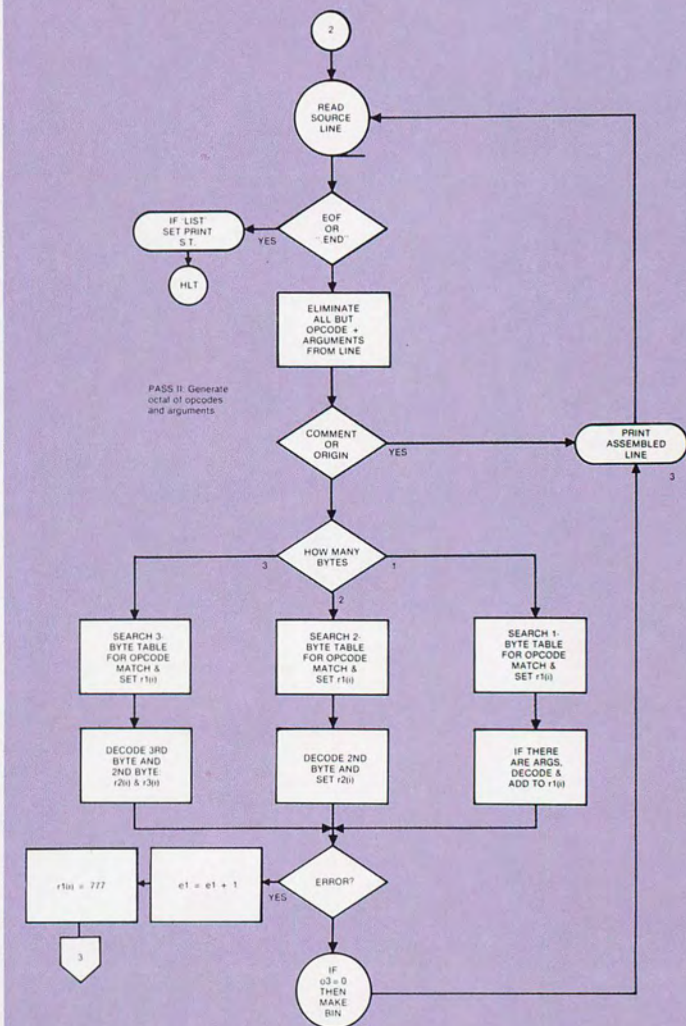
```
OPTIONS? octal-lf-nobin-list
  adrs    op     b2    b3
   0      0      0     0     /
   0      0      0     0     /
   0      0      0     0     /Here's a do-nothing
   0      0      0     0     /sample output from
   0      0      0     0     /the prog CROSS.
   0      0      0     0          *1
   1      61     0     4     start: lxi sp'stack          /set stack
? 4       777    100   0            lbi h'buffer          /set buffer
   7      176    0     0     strt1: mov a'm
                             /Note that CROSS automatically
                             /sets the spacing of the output.
                             /The tabs were not in the source.
   10     315    26    0            call write
   13     176    0     0            mov a'm               /finished?
   14     376    166   0            cpi 166#o
   16     302    22    0            jnz strt2             /nope
   21     166    0     0            hlt                   /yes
   22     16     11    0     strt2: mvi c'i$              /ascii char
   24     166    0     0            hlt
                             /
                             /
                             stack = 2000#o
                             buffer = 100#o
                             write = 26
                             /
                             /
start = 1
strt1 = 7
strt2 = 22
buffer = 100
stack = 2000
write = 26

                             # ERRORS = 1
```

```
options:   ?show
/
/Here's a simple do-nothing
/example of the program.
/
*7#d
start: lxi sp'61#d          /label line
xra a
mov b'a /"'" is a separator
ok: lxi h'load   /fwd ref
mov c'b
write: mvi c'44#o           /octal num
lxi h'load
xra j   /note error code
mvi c'$j          /alpha-num conversion
push d
end: hlt
load = 10#o          /equate line
.end
ox = 32#d  /equate line
jmp write
end: hlt
.end
options:   ?end
```

Figure 1.



PASS I: Each line of source is checked to see if it is a comment, origin, label, or opcode line. Address pointer array "a(i)" is set accordingly, and the symbol table (S.T.) is updated.



PASS II: Generate octal of opcodes and arguments

PROGRAM LISTING

```
10rem************************************************************
20rem*                                                          *
30rem*  A BASIC LANGUAGE CROSS-ASSEMBLER FOR THE INTEL 8080 UP  *
40rem*                                                          *
50rem*                   by P. Reece, 1977                      *
60rem*                                                          *
70rem************************************************************
80rem
90rem
100rem ******************
110rem * BASIC VERSION: *
120rem ******************
130rem PDP-10 Basic is used.  Functions are as follows:
140rem 1) n=instr(a$,b$).  This  returns n equal to the
150rem position of the 1st letter of a$ which matches the
160rem string b$.  Eg. n=instr("abcd","bc") returns n=2.
170rem If no match is found, n is returned as zero.
180rem 2) n$=mid$(a$,a,b).  N$ is returned equal to the substring
190rem of a$ begining with character a, and extending for b characters.
200rem Eg. n$=mid$("abcde",3,2) returns n$="de".
210rem 3) n$=chr$(a).  Converts the integer a to the ascii string
220rem n$. Eg. n$=chr$(56) returns n$="A" (PDP-10 ascii).
230rem 4) n$=str$(a).  Converts the integer a to its ascii equivelent.
240rem Eg. n$=str$(12) returns n$="12".
250rem 5)n=val(a$). Converts the integer ascii a$ to an integer.
260rem Eg., if a$="12", then n is returned equal to 12.
270rem NOTE: The array a(*) must be set equal to the number of statements
280rem in the source.  For most programs, this is generally not more than
290rem 500 statements, so a(500) is sufficient. If an output file is to be
300rem gnerated (via the 'bin' command), the dimensions on r1(*), r2(*), and
310rem r3(*) must also be set.  Otherwise, they may be left small if only a
320rem listing and assembly is desired.
330rem ************
340rem * COMMANDS *
350rem ************
360rem hex:     produces hex listing
370rem octal:   produces octal listing (default)
380rem lf:      prints 7 lf's every 66 lines of listing
390rem nolf:    cancels lf (default)
400rem nolist:  suppresses listing
410rem list:    gives listing (default)
420rem nobin:   suppresses output file
430rem bin:     creates output file (default)
440rem symbol:  prints symbol table of pass#1, then stops
450rem make:    create a source file
460rem .end:    terminate a source file (used during make)
470rem end:     terminate the program
480rem show:    list the unassembled source
490rem **********************
500rem * SPECIAL CHARACTERS *
510rem **********************
520rem $        the next char is ascii & will be translated to octal
530rem :        a label preceeds a colon
540rem #        follows a numeric
550rem d        follows a '#' if the numeric is decimal
560rem o        follows a '#' if the numeric is octal
570rem h        follows a '#' if the numeric is hex
580rem *        preceeds a numeric which is to be used as an origin
590rem /        preceeds a comment
600rem =        indicates an equate line of source
610rem ?        indicates an error in assemblage
620rem **       indicates a hex number too large for the listing
630rem 48      with a '?' for a hex listing, indicates the error byte
640rem 777      as above for '48', but for an octal listing
650rem************************
660rem*     OPTIONS          *
670rem************************
680rem Options in effect if switches = 1:
690rem 01:      hex listing
700rem o2:      suppress listing
710rem o3:      suppress binary output
720rem o4:      generate 7 LFs every 66 lines
730rem o5:      list symbol table, then stop
740rem o6:      'make' a source file
750rem*************************
760rem*    DEFINITIONS        *
770rem*************************
780rem r$(*)=   registers
790rem r1(*)=   byte #1
800rem r2(*)=   byte #2
810rem r3(*)=   byte #3
820rem a(*)=    decimal address of each source line
830rem s$(*)=   symbol table
840rem s1(*)=   octal of s$(*)
850rem m$(*)=   binary of dec 1 -> 7
860rem h2$(*)=  array of hex digits
870rem b1=      pointer for s$(*),s1(*)
880rem b=       pointer for a(*)
890rem e1=      number of assemble errors
900rem***********************************
910rem There are two files - a source file, and an output file.
920rem The output file will contain the assembled octal of the source
930rem in the format: address-1st byte-2nd byte-3rd byte-carriage return.
940rem The address is 6 digits, the bytes are each 4 digits long.
950rem All files are sequentially read (i.e. no direct access is used)
960rem to ensure that non-disc systems can also use the program  (i.e.
970rem a cassette oriented Basic could also use the cross-assembler).
980rem*          INITIALIZE             *
990rem***********************************
1000files s8080.bas$80,o8080.bas$80
1010dimr$(8),r1(100),r2(100),r3(100)
1020dima(200),s$(50),s1(50),m$(8)
1030dimh2$(16)
1040fori=0to9
1050h2$(i)=str$(i)
1060nexti
1070fori=10to15
1080h2$(i)=chr$(i+55)
1090nexti
1100e1=0
1110r$(0)="b"
1120r$(1)="c"
1130r$(2)="d"
1140r$(3)="e"
1150r$(4)="h"
1160r$(5)="l"
1170r$(6)="m"
1180r$(7)="a"
1190m$(0)="000"
1200m$(1)="001"
1210m$(2)="010"
1220m$(3)="011"
1230m$(4)="100"
1240m$(5)="101"
1250m$(6)="110"
1260m$(7)="111"
1270b=0
1280a(0)=0
1290b1=0
1300print"options: ";
1310inputc$
1320ifinstr(c$,"show")<>0then6750
1330ifinstr(c$,"nolist")=0then1350
1340o2=1
1350ifinstr(c$,"nobin")=0then1370
1360o3=1
1370ifinstr(c$,"lf")=0then1390
1380o4=1
1390ifinstr(c$,"symbol")=0then1410
1400o5=1
1410rem
1420print" "
1430ifinstr(c$,"hex")=0then1450
1440o1=1
1450ifc$="make"then6380
1460rem**********************************
1470rem* CREATE ADDRESSES & SYMBOLS      *
1480rem**********************************
1490rem
1500rem Pass one creates a symbol table, and
1510rem calculates all addresses for every
1520rem line of source.
1530rem
1540q=1
1550ifend:1then2290
1560input:1,c$
1570ifinstr(c$,".end")<>0then2290
1580rem comment line only?
1590n1=instr(c$,"/")
1600ifn1=0then1690
1610ifn1>3then1690
1620b=b+1
1630ifq<2then1660
1640a(b)=a(b-1)+q-1
1650goto1530
1660a(b)=a(b-1)
1670goto1530
1680rem origin line?
1690n=instr(c$,"*")
1700ifn=0then1810
1710c$=mid$(c$,n+1)
1720b=b+1
1730k$=c$
1740gosub4770
1750ifk=777then1770
1760c$=str$(k)
1770a(b)=val(c$)
1780a(b)=a(b)-1
1790goto1530
1800rem equate line?
1810n=instr(c$,"=")
1820ifn=0then1920
1830b1=b1+1
1840s$(b1)=mid$(c$,1,n-1)
1850c$=mid$(c$,n+1)
1860k$=c$
1870gosub4770
1880s1(b1)=k
1890b=b+1
1900a(b)=a(b-1)
1910goto1530
1920rem label line?
1930n2=instr(c$,":")
1940ifn2=0then2030
1950b=b+1
1960a(b)=a(b-1)+q
1970b1=b1+1
1980s$(b1)=mid$(c$,1,n2-1)
1990k=a(b)
2000gosub5340
2010s1(b1)=k
2020goto2060
2030rem opcode line? - get the number of bytes
2040b=b+1
2050a(b)=a(b-1)+q
2060ifn1>n2then2080
2070n1=80
2080k$=mid$(c$,n2+1,n1-1)
2090ifinstr(k$,"lxi")=0then2120
2100q=3
2110goto1550
2120rem
2130ifinstr(mid$(k$,1,3),"j")<>0then2100
2140ifinstr(k$,"pchl")<>0then2100
2150ifinstr(k$,"lda")<>0then2100
2160ifinstr(k$,"sta")<>0then2100
2170ifinstr(k$,"hld")<>0then2100
2180ifinstr(k$,"call")<>0then2100
2190ifinstr(k$,"out")=0then2220
2200q=2
2210goto1550
2220rem
2230ifinstr(k$,"inx")<>0then1530
```

```
2240ifinstr(k$,"cpi")<>0then2200
2250ifinstr(k$,"inr")<>0then1530
2260ifinstr(k$,"in")<>0then2200
2270ifinstr(k$,"i")<>0then2200
2280goto1530
2290rem
2300rem*********************************
2310rem* DECODE OPCODES & VARIABLES  *
2320rem*********************************
2330rem
2340rem Pass two decodes opcodes and their
2350rem variables.  The results are stored
2360rem in arrays representing the
2370rem first, second, and third bytes of
2380rem the source line.  The stored values
2390rem are octal.
2400ifo5=1then2840
2410b=0
2420printtab(4);"adrs";
2430printtab(12);"op";
2440printtab(17);"b2";
2450printtab(21);"b3"
2460restore:1
2470goto2510
2480ifend:1then2840
2490rem go print the result for this source line
2500gosub3060
2510input:1,c$
2520e$=c$
2530ifc$=".end"then2840
2540b=b+1
2550rem let k$= opcode + variables only
2560ifinstr(c$,"*")<>0then2480
2570ifinstr(c$,"=")<>0then2480
2580n=instr(c$,"/")
2590ifn=0then2620
2600ifn<6then2480
2610c$=mid$(c$.1.n-1)
2620n=instr(c$,":")
2630ifn=0then2650
2640c$=mid$(c$,n+1)
2650k$=c$
2660rem*****************
2670rem* DECODE 'MOV'  *
2680rem*****************
2690n=instr(c$,"mov")
2700ifn=0then3480
2710k$=mid$(c$,n+4,1)
2720q=n
2730gosub4630
2740ifk=777then2810
2750l=k
2760k$=mid$(c$,q+6,1)
2770gosub4630
2780ifk=777then2810
2790k$="1"+str$(l)+str$(k)
2800goto2820
2810k$="777"
2820r1(b)=val(k$)
2830goto2480
2840rem end of decode phase
2850rem print the symbol table
2860print" "
2870forq=1tob1
2880ifo1<>1then2940
2890k$=str$(s1(q))
2900gosub5820
2910gosub5950
2920prints$(q);"=";k$
2930goto2950
2940prints$(q);"=";s1(q)
2950nextq
2960print" "
2970ife1<>0then3000
2980printtab(10);"NO ERRORS"
2990goto3020
3000printtab(10);"# ERRORS = ";
3010printe1
3020stop
3030rem*********************************
3040rem*  PRINT THE RESULTS            *
3050rem*********************************
3060i=b
3070k=a(i)
3080ifk>0then3110
3090a(i)=0
3100goto3150
3110gosub5340
3120i=b
3130a(i)=k
3140ifa(i)=a(i-1)then3330
3150ifr1(i)>377then3170
3160ifr2(i)<400then3200
3170print"?";
3180e1=e1+1
3190r2(i)=88
3200ifo4<>1then3250
3210t1=t1+1
3220ift1<66then3250
3230t1=0
3240print" "
3250gosub6470
3260ifo2=1then3460
3270ifo1=1then6100
3280rem
3290printtab(4);a(i);
3300printtab(12);r1(i);
3310printtab(17);r2(i);
3320printtab(22);r3(i);
3330n=instr(e$,"/")
3340n1=instr(e$,":")
3350ifn1=0then3400
3360k$=mid$(e$,1,n1)
3370e$=mid$(e$,n1+2)
3380printtab(30);k$;
3390goto3420
3400ifn=0then3420
3410ifn<3then3450
3420printtab(37);e$
3430goto3460
3440printtab(30);e$
3450printtab(30);e$
3460return
3470rem****************
3480rem* DECODE 'MVI' *
3490rem****************
3500n=instr(c$,"mvi")
3510ifn=0then3620
3520k$=mid$(c$,n+4,1)
3530u=n
3540gosub4630
3550r1(b)=15+k
3560n=u
3570k$=mid$(c$,n+6)
3580rem see if k$ is in s.t.
3590gosub4760
3600r2(b)=k
3610goto2480
3620rem*******************
3630rem* DECODE 2 BYTE OP *
3640rem*******************
3650restore
3660data in ,333,out ,323,adi,306,aci,316,sui,326
3670data sbi,336,ani,346,xri,356,ori,366,cpi,376
3680data end,0
3690readk$,k
3700ifk$="end"then3810
3710rem
3720n=instr(c$,k$)
3730ifn=0then3690
3740r1(b)=k
3750l=len(k$)
3760k$=mid$(c$,n+l)
3770c$=k$
3780gosub4760
3790r2(b)=k
3800goto2480
3810rem******************
3820rem* DECODE 'LXI'    *
3830rem******************
3840data lxi b,1,lxi d,21,lxi h,41,lxi sp,61
3850read k$,k
3860ifk$="end"then3980
3870n=instr(c$,k$)
3880ifn=0then3850
3890r1(b)=k
3900k$=c$
3910gosub4760
3920gosub5110
3930goto2480
3940data jnz,302,jz,312,jnz,322,jc,332,jpo,342,jpe,352
3950data jmp,303,jm,372,jp,362,cnz,304,cnc,324,cz,314
3960data cc,334,cpo,344,cpe,354,cp,364,cm ,374,call,315
3970data sta,062,lda,072,shld,042,lhld,052,end,0
3980rem**********************
3990rem* DECODE SINGLE BYTE   *
4000REM**********************
4010data rnz,300,rz,310,rnc,320,rc,330,rpo,340,rpe,350
4020data rp,360,rm,370,ret,311,rlc,7,rrc,17,ral,27,rar,37
4030data xchg,353,xthl,343,sphl,371,pchl,351,hlt,166,nop,0
4040data di,363,ei,373,daa,47,cma,57,stc,67,cmc,77,end,0
4050readk$,k
4060ifk$="end"then4100
4070ifinstr(c$,k$)=0then4050
4080r1(b)=k
4090goto2480
4100rem see if line is a one byte + register instr
4110data pop,301,push,305,stax,2,ldax,12
4120data inx,3,dcx,13,dad,11,end,0
4130readk$,k
4140ifk$="end"then4290
4150n=instr(c$,k$)
4160ifn=0then4130
4170l=len(k$)
4180k$=mid$(c$,n+l+1,1)
4190n=instr("bdh",k$)
4200ifn=0then4240
4210k=k+(n-1)*20
4220r1(b)=k
4230goto2480
4240ifk$<>"s"then4270
4250k=k+60
4260goto4220
4270ifk$="p"then4250
4280goto4530
4290rem decode final single byte opcode
4300data add,20,adc,21,sub,22,sbb,23
4310data ana,24,xra,25,ora,26,cmp,27,end,0
4320readk$,a
4330ifk$="end"then4410
4340n=instr(c$,k$)
4350ifn=0then4320
4360k$=mid$(c$,n+4,1)
4370gosub4630
4380r1(b)=a*10+k
4390goto2480
4400data inr,4,dcr,5,end,0
4410readk$,a
4420ifk$="end"then4500
4430n=instr(c$,k$)
4440ifn=0then4410
4450k$=mid$(c$,n+4,1)
4460gosub4630
4470ifk=777then4510
4480r1(b)=k*10+a
4490goto2480
4500rem*******************
4510rem*     ERROR       *
4520rem*******************
4530r1(b)=777
```

```
4540goto2480
4550rem
4560rem
4570rem
4580rem               SUBROUTINES
4590rem               -----------
4600rem
4610rem
4620rem
4630rem***************************
4640rem*  DECODE A REGISTER      *
4650rem***************************
4660rem enter with:     k$= register
4670rem exit with:      k= octal of register
4680rem                 k= 777 if error
4690k=777
4700fori=0to7
4710ifk$<>r$(i)then4740
4720k=i
4730i=7
4740nexti
4750return
4760rem*************************
4770rem* DECODE  #,ASCII,S.T.  *
4780rem*************************
4790rem enter with: k$ = string for decoding
4800rem exit with:  k = octal if s.t. entry
4810rem             k = octal if ascii entry
4820rem             k = octal if numeric entry
4830rem             k = 777 if none of the above
4840k=777
4850a$=k$
4860k$=c$
4870n=instr(k$,"´")
4880ifn=0then4900
4890k$=mid$(k$,n+1)
4900n=instr(k$,"#")
4910ifn=0then4980
4920rem reach here if a numeral existed
4930b$=mid$(k$,n+1,1)
4940k=val(mid$(k$,1,n-1))
4950ifb$="o"then5100
4960gosub5330
4970goto5100
4980rem see if k$ is ascii
4990n=instr(a$,"$")
5000if n=0then5040
5010k=instr("abcdefghijklmnopqrstuvwxyz",mid$(a$,n+1,1))
5020gosub5330
5030goto5100
5040rem see if a$ is in the symbol table
5050fori=1tob1
5060ifinstr(a$,s$(i))<>0then5090
5070nexti
5080goto5100
5090k=s1(i)
5100return
5110rem*************************
5120rem*   DECODE BYTE #3      *
5130rem*************************
5140rem enter with:  k = octal
5150rem exit with: r2(b) = octal if k<377
5160rem          : r3(b) = octal if k>377
5170r2(b)=k
5180ifk>377then5200
5190goto5320
5200rem convert k to a binary string
5210k$=str$(k)
5220gosub5460
5230h$=mid$(k$,1,8)
5240l$=mid$(k$,9)
5250rem convert h & l to octal

5260k$=h$
5270gosub5640
5280r3(b)=val(k$)
5290k$=l$
5300gosub5640
5310r2(b)=val(k$)
5320return
5330rem**********************************
5340rem* CONVERT DECIMAL K TO OCTAL K *
5350rem**********************************
5360a1=k
5370h$=""
5380fori=4to0by-1
5390a=int(a1/8**i)
5400k1=a1-a*8**i
5410a1=k1
5420h$=h$+str$(a)
5430nexti
5440k=val(h$)
5450return
5460rem***************************************
5470rem*   CONVERT OCTAL K$ TO BINARY K$   *
5480rem***************************************
5490h$=""
5500fori=1tolen(k$)
5510k2$=mid$(k$,i,1)
5520fork=0to7
5530ifstr$(k)<>k2$then5560
5540h$=h$+m$(k)
5550k=8
5560nextk
5570nexti
5580k$=h$
5590rem make k$ 16 characters long
5600fori=1to16-len(k$)
5610k$="0"+k$
5620nexti
5630return
5640rem*************************************
5650rem*   MAKE BINARY K$ OCTAL K$        *
5660rem*************************************
5670rem enter with:  k$ = 8-char binary
5680rem exit with:   k$ = octal equivilant
5690k$="0"+k$
5700k3$=""
5710fori=1to9step3
5720k2$=mid$(k$,i,3)
5730fork=0to7
5740ifinstr(k2$,m$(k))=0then5770
5750k3$=k3$+str$(k)
5760k=8
5770nextk
5780nexti
5790k$=k3$
5800return
5810rem****************************************
5820rem*  CONVERT OCTAL K$ TO DECIMAL K$   *
5830rem****************************************
5840k=0
5850l=len(k$)
5860l1=l
5870l=l+1
5880fori=1tol1
5890l=l-1
5900k=k+val(mid$(k$,l,1))*8**(i-1)
5910nexti

5920k$=str$(k)
5930return
5940rem****************************************
5950rem*  CONVERT DECIMAL K$ TO HEX K$     *
5960rem****************************************
5970a1=val(k$)
5980h1$=""
5990fori=n7to0step-1
6000k=int(a1/16**i)
6010k1=a1-k*16**i
6020ifk<17then6050
6030h1$="**"
6040goto6080
6050h1$=h1$+h2$(k)
6060a1=k1
6070nexti
6080k$=h1$
6090return
6100rem*************************
6110rem*  OUTPUT A HEX LINE   *
6120rem*************************
6130k$=str$(a(i))
6140n7=4
6150gosub5820
6160gosub5950
6170printtab(4);k$;
6180k$=str$(r1(b))
6190n7=1
6200gosub5820
6210gosub5950
6220ifk$<>"00"then6240
6230k$="  "
6240printtab(12);k$;
6250k$=str$(r2(b))
6260gosub5820
6270gosub5950
6280ifk$<>"00"then6300
6290k$="  "
6300printtab(17);k$;
6310k$=str$(r3(b))
6320gosub5820
6330gosub5950
6340ifk$<>"00"then6360
6350k$="  "
6360printtab(21);k$;
6370goto3330
6380rem*************************************
6390rem*        MAKE COMMAND              *
6400rem*************************************
6410inputc$
6420ifc$<>".end"then6450
6430print:1,c$
6440stop
6450print:1,c$
6460goto6410
6470rem***************************************
6480rem*    CREATE OCTAL OUTPUT FILE       *
6490rem***************************************
6500k$=str$(a(i))
6510gosub6640
6520k1$=k$
6530k$=str$(r1(i))
6540gosub6640
6550k1$=k1$+k$
6560k$=str$(r2(i))
6570gosub6640
6580k1$=k1$+k$
6590k$=str$(r3(i))
6600gosub6640
6610k1$=k1$+k$
6620print:2,k1$
6630return
6640rem***************************
6650rem* MAKE K$ = 3 CHARACTERS  *
6660REM***************************
6670n=len(k$)
6680ifn=3then6730
6690ifn=2then6720
6700k$="00"+k$
6710goto6730
6720k$="0"+k$
6730return
6740rem *******
6750rem *SHOW *
6760rem *******
6770ifend:1then6810
6780input:1,c$
6790printc$
6800goto6770
6810restore:1
6820goto1300
6830end

READY
```

STRUCTURED PROGRAMMING

Lately the computer literature has been full of articles on Structured Programming and its advantages. It has been suggested that instead of making flowcharts we write our programs using the structured blocks and pseudocodes and then *translate* these blocks into their equivalent in BASIC. But why should we have to do the translation? Should we not make the computer do this task?

This program does exactly this. It lets you use the IFTHEN-ELSE, DOUNTIL and DOWHILE structured sequences and the pseudocodes associated with them, along with standard BASIC statements in your programs. It will read your program statement by statement and convert it into standard BASIC statements. This sort of a translator is called a "preprocessor."

The following examples show how the structured sequences are changed into BASIC.

DOUNTIL BLOCK (See Figure 1)	**IN BASIC**
10 DO	10 REM DO
20 . . .	20 . . .
30 . . .	30 . . .
40 UNTIL X=5	40 IF X<>5 THEN 10
DOWHILE BLOCK (See Figure 2)	
10 WHILE X=5	10 IF X<>5 THEN 50
20 . . .	20 . . .
30 . . .	30 . . .
40 ENDWHILE	40 GOTO 10
50 . . .	50 . . .
IFTHENELSE BLOCK (See Figure 3)	
10 REM IF X=5 THEN	10 IF X<>5 THEN 50
20 . . .	20 . . .
30 . . .	30 . . .
40 ELSE	40 GOTO 70
50 . . .	50 . . .
60 . . .	60 . . .
70 ENDIF	70 REM ENDIF

Note that when a structured IF is used it has to be preceded by a REM so that it can be distinguished from the IF of BASIC, and also that the THEN of the structured IF is not followed by anything else. Another thing to be observed is that the expression in the pseudocode is always complemented during the translation because the IF of BASIC takes a jump if the expression in it is true, while in the structured IF it is just the opposite.

To use this BASIC preprocessor enter your program statement by statement, including the structured pseudocodes as if they were ordinary BASIC statements (preceded by line numbers), but remember to type a REM before using a structured IF. To end the program type BASIC's END statement.

The preprocessor looks at every statement and if it is a standard BASIC statement it outputs the line immediately without any change. In the BASIC I use, the PRINT ON (2) statement (e.g. Line 590 in the listing) is used to output a line onto th epaper tape punch. This will require modification in other BASIC interpreters. The pseudocodes have to be processed before they are punched out. Some (e.g. DO,UNTIL) are converted and punched out immediately after their input — others like WHILE can be punched out only at a later stage; in the case of WHILE only after the statement after ENDWHILE has been read. So after you have typed a statement, if there is no output on the punch, then type the next statement.

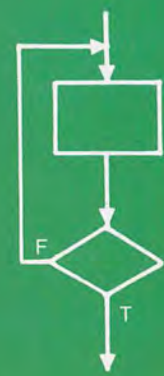


Figure 1. DOUNTIL

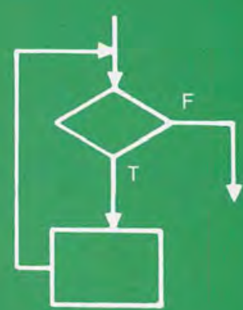


Figure 2. DOWHILE

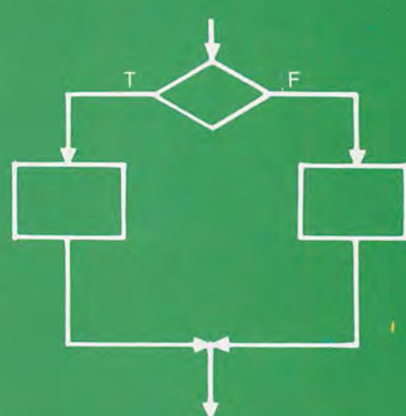


Figure 3. IF-THEN-ELSE

IN BASIC

By Kamal Shah

Lines containing commas should be entered within quotes. Otherwise, as commas separate two variables, the input will be considered as being made up of two or more variables while really only one is expected (Line 310 in the listing).

The statements that require processing are listed below along with what is done in each case:

DO (Lines 930-990)
1) Stack the line number in the DOLINENUMBER(F$) stack.
2) Insert a 'REM' between the line number and the 'DO'.
3) Output the converted line.
4) Go to read the next line.

UNTIL (Lines 1010-1140)
1) Check if the DOLINENUMBER stack is empty. If it is so then print error (because there is no corresponding DO statement) and go to step 4.
2) If there is a line number then unstack it; get the expression in the UNTIL statement and complement it.
3) Output the present line number followed by 'IF' + the complemented expression + 'GOTO' + the line number unstacked in step 2.
4) Go to read the next line.

WHILE (Lines 1160-1220)
1) Stack the line number in the WHILELINENUMBER(Y$) stack.
2) Stack the expression in the WHILELINE(W$) stack.

ENDWHILE (Lines 1240-1350)
1) Check if the WHILELIENNUMBER stack is empty. If it is so then print error and go to step 5 (note that the ENDWHILEFOUND(F1) flag is not set).
2) If there is a line number then unstack the line number at the top of the WHILELINENUMBER stack.
3) Output the present line number followed by 'GOTO' + the line number unstacked in step 2.
4) Set the ENDWHILEFOUND flag.
5) Go to read the next line.

Whenever a statement is read and the ENDWHILEFOUND flag is found to be set, the following occurs before the line read is processed: (Lines 490-610)

a. Unstack the expression in the WHILELINE stack and complement it.
b. Output the line number of the WHILELINE followed by 'IF' + the complemented expression + 'GOTO' + the present line number.
c. Reset the ENDWHILEFOUND flag.

REM
1) As soon as a REM statement is recognized, check if it is followed by an IF or not. (Lines 1370-1420)
2) If it is not then just output the statement as it is and go on to read the next line. (Lines 1510-1530)

Structured IF (Lines 1430-1500)
1) Stack the expression and the rest of the line in the IFLINE(I$) stack, and the line number in the IFLINENUMBER(J$) stack.
2) Go to read the next line.

ELSE (Lines 1550-1600)
1) Stack the line number in the ELSELINENUMBER(G$) stack.
2) Set the ELSEFOUND(F2) flag.

Whenever a statement is read and the ELSEFOUND flag is set then before the line read is processed the following occurs: (Lines 630-780)

a. Check if the IFLINENUMBER stack is empty. If it is, then print error and go to step d.
b. If there is a line number then unstack it; unstack also from the IFLINE stack and complement the expression in it.
c. Output the unstacked line number followed by 'IF' + the complemented expression + the rest of the IFLINE + the present line number.
d. Reset the ELSEFOUND flag.

ENDIF (Lines 1620-1750)
1) Check if the ELSELINENUMBER stack is empty. If it is so then print error and read the next line.
2) Otherwise unstack the line number from the ELSELINENUMBER stack, add to it 'GOTO' + the present line number and output it.
3) Output the present line number + 'REM ENDIF'.

END (Lines 800-910)
1) Check if any of the DOLINENUMBER, WHILELINENUMBER or IFLINENUMBER stacks are not empty — if so then print error and set the ENDFOUND(E2) flag; if there is nothing remaining in the stacks then just set the ENDFOUND flag.

There are two subroutines in this program:
1) The subroutine at Line 1820 receives from the main program an expression in C$. It copies C$ into E$ until it finds a relational operator (=,< >,<=,>=,<,>) in C$. It converts the operator found in C$ into its complement (< >,=,>,<,>=,<=) and places that and the rest of C$ into E$ and returns.
2) The subroutine at Line 2220 looks at A$ (the statement that has been read by the main program), skips all the initial blanks, find out the line number, copies the line number into L$ and also sends out the LINENUMBERFOUND(Z9) flag to indicate if a line number was found or not.

The following are the error messages printed along with their meanings.

DO ERROR:	more DO	lines than UNTIL	lines
ERROR DO:	more UNTIL	lines than DO	lines
WHILE ERROR:	more WHILE	lines than ENDWHILE	lines
ERROR WHILE:	more ENDWHILE	lines than WHILE	lines
IF ERROR:	more IF	lines than ELSE	lines
ERROR IF:	more ELSE	lines than IF	lines
or	more ENDIF	lines than ELSE	lines
NO LINE NUMBER:	the line that was input last did not have a line number.		

The maximum number of nestings permitted for any particular structured sequence is ten. To alter this number, change the dimension of F$(for DOUNTIL), W$ and Y$(for DOWHILE) and I$, J$ and G$(for IF-THENELSE) in the lines 30 and 40. To reduce and memory required to store this preprocessor remove all the REM statements.

In the past I had written some programs using structured program sequences, then translated them by hand into BASIC. I found that the debugging time for lone programs reduces by a factor of two and often more. As this translation by hand was quite time-consuming I found myself not using structured sequences despite their ob-

ELECTRONIC SYSTEMS
p.o. box 212 Burlingame CA 94010
(408) 374-5984

UART & BAUD RATE GENERATOR

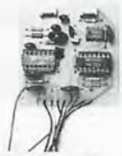
Part no. 101
- Converts serial to parallel and parallel to serial
- Low cost on board baud rate generator
- Baud rates: 110, 150, 300, 600, 1200, and 2400
- Low power drain +5 volts and -12 volts required
- TTL compatible
- All characters contain a start bit, 5 to 8 data bits, 1 or 2 stop bits, and either odd or even parity.
- All connections go to a 44 pin gold plated edge connector
- Board only $12.00; with parts $35.00

RS-232/TTL INTERFACE

Part no. 232
- Converts TTL to RS-232, and converts RS-232 to TTL
- Two separate circuits
- Requires -12 and +12 volts
- All connections go to a 10 pin gold plated edge connector
- Board only $4.50; with parts $7.00

DC POWER SUPPLY

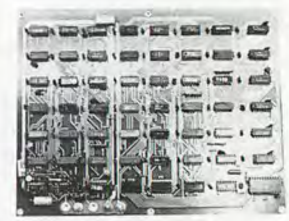
Part no. 6085
- Board supplies a regulated +5 volts at 3 amps., +12,-12, and -5 volts at 1 amp.
- Board has filters, rectifiers, and regulators
- Power required is 8 volts AC at 3 amps., and 24 volts AC C.T. at 1.5 amps.
- Board only $12.50



TAPE INTERFACE
Part no. 111
- Play and record Kansas City Standard tapes
- Converts a low cost tape recorder to a digital recorder
- Works up to 1200 baud
- Digital in and out are TTL-serial
- Output of board connects to mic. in of recorder
- Earphone of recorder connects to input on board
- Requires +5 volts, low power drain
- Board $7.60; with parts $27.50
- No coils

Part no. 107
RF MODULATOR
- Converts video to AM modulated RF, Channels 2 or 3
- Power required is 12 volts AC C.T., or +5 volts DC
- Board $4.50; with parts $13.50

TELEVISION TYPEWRITER



Part no. 106
- Stand alone TVT
- 32 char/line, 16 lines, modifications for 64 char/line included
- Parallel ASCII (TTL) input
- Video output
- 1K on board memory
- Output for computer controlled curser
- Auto scroll
- Non-distructive curser
- Curser inputs: up, down, left, right, home, EOL, EOS
- Scroll up, down
- Requires +5 volts at 1.5 amps, and -12 volts at 30 mA
- Board only $39.00; with parts $145.00

8K STATIC RAM

Part no. 300
- 8K Altair bus memory
- Uses 2102 Static memory chips
- Memory protect
- Gold contacts
- Wait states
- On board regulator
- S-100 bus compatible
- Vector input option
- TRI state buffered
- Board only $22.50; with parts $160.00

TIDMA
Part no. 112
- Tape Interface Direct Memory Access
- Record and play programs without bootstrap loader (no prom) has FSK encoder/decoder for direct connections to low cost recorder at 625 baud rate, and direct connections for inputs and outputs to a digital recorder at any baud rate.
- S-100 bus compatible
- Comes assembled and tested for $160.00

APPLE I MOTHER BOARD

Part no. 102
- 10 slots – 44 pin (.156) connectors spaced ¾" apart
- Connects to edge connector of computer
- Pin 20 and 22 connects to X & Z for power and ground
- Board has provisions for bypass capacitors
- Board cost $15.00

MODEM

Part no. 109
- Type 103
- Full or half duplex
- Works up to 300 baud
- Originate or Answer
- No coils, only low cost components
- TTL input and output-serial
- Connect 8 ohm speaker and crystal mic. directly to board
- Uses XR FSK demodulator
- Requires +5 volts
- Board $7.60; with parts $27.50

To Order:

VISA master charge

Mention part number and description. For parts kits add "A" to part number. Shipping paid for orders accompanied by check, money order, or Master Charge, BankAmericard, or VISA number, expiration date and signature. Shipping charges added to C.O.D. orders. California residents add 6.5% for tax. Parts kits include sockets for all ICs, components, and circuit board. Documentation is included with all products. Dealer inquiries invited. 24 Hour Order Line: (408) 374-5984. E21

vious advantages. Now I hope, if there are others who have had the same experience, that they will find this program useful as well as interesting.

PROGRAM LISTING

```
10 REM   BASIC PREPROCESSOR WRITTEN BY KAMAL SHAH AND BIMLA SUREKA.
20 REM   SRI AUROBINDO ASHRAM, PONDICHERRY 605002, INDIA
30 DIM A$(80), B$(8), C$(80), E$(50), F$(6, 10), G$(6, 10)
40 DIM I$(50, 10), J$(6, 10), L$(6), W$(50, 10), Y$(6, 10)
50 PRINT "ENTER YOUR PROGRAM LINE BY LINE. IF THE LINE CONTAINS"
60 PRINT "A COMMA(S), ENTER THE LINE WITHIN QUOTATION MARKS. IF"
70 PRINT "THE LINE IS A STRUCTURED 'IF' STATEMENT THEN TYPE 'REM'"
80 PRINT "BETWEEN THE LINE NO. AND THE 'IF'"
90 REM      INITIALIZE ALL FLAGS AND STACKPOINTERS
100 REM      E2=ENDFOUNDFLAG - SET IF "END" IS FOUND
110 REM      F1=ENDWHILEFOUND FLAG - SET IF AN "ENDWHILE" IS FOUND
120 REM      F2=ELSEFOUNDFLAG - SET IF AN "ELSE" IS FOUND
130 REM      D1=STKPTR FOR THE DOLINENUMBER STACK(F$)
140 REM      W1=STKPTR FOR THE WHILELINE STACK(W$)
150 REM      W2=STKPTR FOR THE WHILELINENUMBER STACK(Y$)
160 REM      I1=STKPTR FOR THE IFLINE STACK(I$)
170 REM      I2=STKPTR FOR THE IFLINENUMBER STACK(J$)
180 REM      E1=STKPTR FOR THE ELSELINENUMBER STACK(G$)
190 E2=0
200 F1=0
210 F2=0
220 D1=1
230 W1=1
240 W2=1
250 I1=1
260 E1=1
270 I2=1
280 B$="        "
290 REM      CHECK IF ENDFOUNDFLAG IS SET - IF SO STOP ELSE READ A LINE
300 IF E2<>0 GOTO 1810
310 INPUT A$
320 L=LEN(A$)
330 REM      Z9=LINENUMBERFOUND FLAG - THIS FLAG IS SET TO "1" IF
340 REM      NO LINENUMBER IS FOUND, ELSE IT IS GREATER THAN "1"
350 Z9=1
360 GOSUB 2160
370 IF Z9>1 THEN 400
380 PRINT "NO LINE NUMBER"
390 GOTO 310
400 IF L>10 THEN 450
410 REM      INCREASE THE LENGTH OF THE LINE READ (IF IT IS LESS THAN


420 REM      10) BY ADDING 8 BLANKS TO PREVENT SUBSCRIPT ERRORS
430 REM      WHILE CHECKING FOR "ENDWHILE", "ENDIF" ETC
440 A$=A$+B$
450 IF A$(I, I)<>" " GOTO 480
460 I=I+1
470 GOTO 450
480 IF F1<>1 GOTO 620
490 REM      ENDWHILEFOUND FLAG IS SET. UNSTACK THE PREVIOUS WHILELINE,
500 REM      COMPLEMENT THE EXPRESSION IN IT, TAKE THE LINENUMBER
510 REM      AT THE TOP OF THE WHILELINE NUMBER STACK, CONVERT AND OUTPUT
520 REM      THE WHILE LINE AND RESET THE ENDWHILEFOUND FLAG. PROCESS THE
530 REM      PRESENT LINE
540 W1=W1-1
550 C$=W$(W1)
560 K=1
570 GOSUB 1820
580 C$=Y$(W2)+" IF "+E$+" GOTO "+L$
590 PRINT ON (2)C$
600 F1=0
610 GOTO 790
620 IF F2<>1 GOTO 790
630 REM      ELSEFOUND FLAG IS SET. CHECK FOR AN INFLINE IN THE STACK.
640 REM      IF THERE ISN'T ONE THEN PRINT ERROR, ELSE UNSTACK THE
650 REM      THE PREVIOUS IFLINE, COMPLEMENT THE EXPRESSION IN IT, CONVERT
660 REM      AND OUTPUT THAT LINE AND RESET THE ELSEFOUND FLAG
670 IF I2>1 GOTO 710
680 PRINT "ERROR IF"
690 F2=0
700 GOTO 790
710 I1=I1-1
720 C$=I$(I1)
730 I2=I2-1
740 K=1
750 GOSUB 1820
760 C$=J$(I2)+" IF "+E$+" "+L$
770 PRINT ON (2)C$
780 F2=0
790 IF A$(I, I+3)<>"END " GOTO 920
800 REM      "END" STATEMENT FOUND. CHECK FOR EXTRA "DO", "WHILE", "IF"
810 REM     LINE NUMBERS REMAINING IN THE STACKS. IF SO PRINT ERROR,
820 REM     ELSE CONTINUE. IN BOTH CASES SET THE ENDFOUND FLAG
830 IF D1=1 GOTO 850
840 PRINT "DO ERROR"
850 IF W2=1 GOTO 870
860 PRINT "WHILE ERROR"
870 IF I2=1 GOTO 890
880 PRINT "IF ERROR"
890 PRINT ON (2)A$
900 E2=1
910 GOTO 1800
920 IF A$(I, I+1)<>"DO" GOTO 1000
930 REM      A "DO" STATEMENT. STACK THE LINE NUMBER, INSERT A "REM"
940 REM      BETWEEN THE LINE NUMBER AND "DO" AND OUTPUT THE LINE
950 F$(D1)=L$
960 D1=D1+1
970 A$=L$+" REM DO"
980 PRINT ON (2)A$
990 GOTO 1800
1000 IF A$(I, I+4)<>"UNTIL" GOTO 1150
1010 REM      AN "UNTIL" STATEMENT. CHECK FOR A DOLINE NUMBER IN THE
1020 REM      STACK. IF THERE ISN'T ONE THEN PRINT ERROR, ELSE UNSTACK
1030 REM      THE LINE NUMBER, COMPLEMENT THE EXPRESSION IN THE "UNTIL"
1040 REM      LINE, CONVERT AND OUTPUT IT
1050 IF D1>1 GOTO 1080
1060 PRINT "ERROR DO"
1070 GOTO 1800
1080 K=I+5
1090 C$=A$
1100 D1=D1-1
1110 GOSUB 1820
1120 A$=L$+" IF "+E$+" GOTO "+F$(D1)
1130 PRINT ON (2)A$
1140 GOTO 1800
1150 IF A$(I, I+4)<>"WHILE" GOTO 1230
1160 REM      A "WHILE" STATEMENT. STACK THE LINE NUMBER AND THE EXPRESSIO
1170 REM      IN THE WHILELINE AS THE WHILELINE CANNOT BE CONVERTED NOW.
1180 W$(W1)=A$(I+5, L)
1190 W1=W1+1
1200 Y$(W2)=L$
1210 W2=W2+1
1220 GOTO 1800
1230 IF A$(I, I+7)<>"ENDWHILE" GOTO 1360
1240 REM      AN "ENDWHILE" STATEMENT. CHECK FOR A WHILE LINE IN THE WHILE
1250 REM      LINE STACK. IF THERE ISN'T ONE THEN PRINT ERROR, ELSE GET TH
1260 REM      LINE NUMBER OF THE LAST WHILE STATEMENT, CONVERT AND OUTPUT
1270 REM      THE ENDWHILE STATEMENT AND SET THE ENDWHILEFOUND FLAG.
1280 IF W2>1 GOTO 1310
1290 PRINT "ERROR WHILE"
1300 GOTO 1800
1310 W2=W2-1
1320 A$=L$+" GOTO "+Y$(W2)
1330 F1=1
1340 PRINT ON (2)A$
1350 GOTO 1800
1360 IF A$(I, I+2)<>"REM" GOTO 1540
1370 REM      A "REM" STATEMENT
1380 I=I+3
1390 IF A$(I, I)<>" " GOTO 1420
1400 I=I+1
1410 GOTO 1390
1420 IF A$(I, I+1)<>"IF" GOTO 1510
1430 REM      "REM" IS FOLLOWED BY A STRUCTURED "IF" STATEMENT. AS THIS
1440 REM      "IF" CANNOT BE CONVERTED NOW, STACK THE EXPRESSION AND
1450 REM      REST OF THE "IF" LINE AND ITS LINE NUMBER
1460 I$(I1)=A$(I+2, L)
1470 I1=I1+1
1480 J$(I2)=L$
1490 I2=I2+1
1500 GOTO 1530
1510 REM      IT IS AN ORDINARY "REM" STATEMENT SO JUST OUTPUT IT
1520 PRINT ON (2)A$
1530 GOTO 1800
1540 IF A$(I, I+3)<>"ELSE" GOTO 1610
1550 REM      AN "ELSE" STATEMENT. IT CANNOT BE CONVERTED NOW SO STACK
1560 REM      THE LINE NUMBER AND SET THE ELSEFOUND FLAG
1570 G$(E1)=L$
1580 E1=E1+1
1590 F2=1
1600 GOTO 1800
1610 IF A$(I, I+4)<>"ENDIF" GOTO 1760
1620 REM      AN ENDIF STATEMENT. CHECK FOR AN ELSELINE NUMBER IN THE
1630 REM      STACK. IF THERE ISN'T ONE THEN PRINT ERROR, ELSE UNSTACK THE
1640 REM      LINE NUMBER. ADD TO IT "GOTO"+THE PRESENT LINE NUMBER AND
1650 REM      OUTPUT THE LINE. THEN INSERT "REM" IN BETWEEN THE
1660 REM      PRESENT LINE NUMBER AND THE "ENDIF" AND OUTPUT THIS ALSO.
1670 IF E1>1 GOTO 1700
1680 PRINT "ERROR IF"
1690 GOTO 1800
1700 E1=E1-1
1710 C$=G$(E1)+" GOTO "+L$
1720 PRINT ON (2)C$
1730 A$=L$+" REM ENDIF"
1740 PRINT ON (2)A$
1750 GOTO 1800
1760 REM      RESTORE THE INPUT STATEMENT TO ITS ORIGINAL SIZE IF
1770 REM      EIGHT BLANKS HAD BEEN ADDED TO IT
1780 A$=A$(1, L)
1790 PRINT ON (2)A$
1800 GOTO 300
1810 STOP
1820 REM      SUBR. COMPLEMENTS EXPRESSION IN C$ AND PUTS IT IN E$
1830 M1=1
1840 REM      "K1" REMAINS ZERO UNTIL A RELATIONAL OPERATOR IS FOUND IN C$
1850 REM    IT TAKES DIFFERENT VALUES ACCORDING TO THE OPERATOR FOUND
1860 K1=0
1870 L1=LEN(C$)
1880 REM      "K" IS THE INDEX FROM THE MAIN PROGRAM INDICATING FROM
1890 REM      WHICH CHARACTER WE HAVE TO SEARCH FOR THE EXPRESSION
1900 IF C$(K, K)<"<" THEN 1940
1910 IF C$(K, K)<=">" THEN 1980
1920 REM    OPERATOR NOT YET FOUND SO COPY A CHARACTER FROM C$
1930 REM    INTO E$ AND GO BACK TO CHECK THE NEXT CHARACTER IN C$
1940 E$(M1, M1)=C$(K, K)
1950 M1=M1+1
1960 K=K+1
1970 GOTO 1900
1980 REM    FOUND AN OPERATOR. FIND OUT ITS COMPLEMENT AND PUT
1990 REM    INTO E$. THE CHARACTERS IN C$ AFTER THE OPERATOR
2000 REM    ARE PUT INTO E$ WITHOUT ANY CHANGE
2010 IF C$(K+1, K+1)="=" THEN 2090
2020 IF C$(K+1, K+1)=">" THEN 2090
2030 IF C$(K, K)="<" THEN E$(M1, M1+1)=">="
2040 IF C$(K, K)=">" THEN E$(M1, M1+1)="<="
2050 IF C$(K, K)="=" THEN E$(M1, M1+1)="<>"
2060 K=K+1
2070 M1=M1+2
2080 GOTO 2140
2090 IF C$(K, K+1)="<=" THEN E$(M1, M1)=">"
2100 IF C$(K, K+1)=">=" THEN E$(M1, M1)="<"
2110 IF C$(K, K+1)="<>" THEN E$(M1, M1)="="
2120 K=K+2
2130 M1=M1+1
2140 E$(M1, M1+L1-K)=C$(K, L1)
2150 RETURN
2160 REM      SUBR. TO FIND LINE NUMBER IN A$
2170 K2=1
2180 REM      SKIP ALL THE INITIAL BLANKS
2190 IF A$(K2, K2)<>" " GOTO 2240
2200 K2=K2+1
2210 GOTO 60
2220 REM      RETURN TO MAIN PROGRAM WHEN A NONBLANK AND NONNUMERIC
2230 REM      CHARACTER IS FOUND.
2240 IF A$(K2, K2)<"0" GOTO 2310
2250 IF A$(K2, K2)>"9" GOTO 2310
2260 L$(Z9, Z9)=A$(K2, K2)
2270 K2=K2+1
2280 Z9=Z9+1
2290 GOTO 2240
2300 REM      I=POINTER TO THE FIRST CHARACTER IN A$ AFTER THE LINE NO.
2310 I=K2
2320 RETURN
2330 END
```

Product Preview
Sord Computer

The SORD Computer Systems Company was formed in 1970. Since then the firm has grown from a small manufacturer of minicomputers into one of Japan's leading producers of small systems.

Its most recent product is the SORD M200, a basic system which consists of a Z-80 CPU, minifloppy diskette, a 12-inch CRT and a discharged type printer. The M200 series includes the M220 standard model (see Photo 1), the M230 Lab, designed for laboratory applications, the M250 CCR, a computerized cash register and the M270 TSS terminal for timesharing applications.

The Z-80 chip in the central processing unit was chosen for its low cost, the efficiency of the machine's language and the availability of developing equipment. Although the system was designed for dedicated professional application, it is hoped that a substantial number will be sold for personal use.

MEMORY CAPACITY

The M200 series has built-in flexible RAMs for the main program storage area. The M220 and the M230 Lab have 32K RAMs while the M250 CCR and the M270 TSS incorporate 16K RAMs. The RAMs permit more comprehensive calculations and data processing than with conventional models. Memory capacity can be boosted up to 48K bytes.

The ROMs contain the keyboard scanning routines, the video display drivers, and the minifloppy diskette IPL routines. The ROMs are of the mask programmable (non-erasable) variety, and total 1-2K bytes according to the product line.

I/O EQUIPMENT

The keyboard is a full size, user-oriented multi-purpose, professional quality 125-key unit. The data loading is simplified by 26 designated keys with four commands. The frequently used keywords and statements in BASIC can be automatically transmitted to the processor by using a specially designated key. There are seven mode keys and five shift keys to produce highly efficient data processing.

Four LEDs help the user prevent operational errors, indicating one mode at a time. Moreover, the intelligent-type M200 series also gives the user a smart capability to enable special keyboard arrangement.

The M200 series uses the CRT video display with a 12-inch flat face Braun tube. The video display logic generates the necessary video and synch signals to form 24 lines of 80 characters. Regular alphanumerics and signals of 1920 and 191 kinds are displayed as a dot matrix.

By Akio Guji and Kazuhiko Nishi

The RAM is accessed by the Z-80 CPU only when information is to be read from or written to the screen. The graphics applied in the M200 series are displayed on a character-by-character basis.

PRINTER

The built-in discharge-type printer can print out a maximum of two lines of 40 characters per second. All of 1,920 characters (categorized in 127 kinds) applied in BASIC of the M200 series can be printed out.

If necessary, an optional 80-digit, 40-65 characters/second printer can be connected. Moreover, other 136-digit or less printers are also applicable.

The M200 series has another flexible feature that permits the combined use of a series of computer system units. Every model incorporates a 4-slot to be applied in S-100 bus, an international mainstay bus, providing a connection with many interfaces now marketed.

One slot is reserved for the memory system on the side of the unit, while three slots for data loading memory. This gives the user voluntary programmable processing.

Furthermore, if the IEEE 488 Interface Bus is connected to the terminal, the M200 series links to variable I/O units, ranging from ordinary terminal units to advanced measuring devices and further graphic/scientific simulators. The IEEE 488 is an optional home and personal interface bus (HP-IB).

An ordinary tape recorder can be easily connected to both the M220 standard model and the M230 Lab model without any optional unit.

The connection to a mini- or large scale computer can be applicable by using an optional MODEM.

Additional optional parts for the M200 series include A/D and D/A converters (the M230 Lab already incorporates), DIDO, ASCII (HP-IB) Bus Converter, Bar Code Reader/Printer, OCR Reader, and more.

STORAGE EQUIPMENT

The M200 series has a built-in minifloppy diskette drive. Compact in size, a 5-inch diskette has a 35-track, 71.6K bytes format capacity. Each track has eight sectors whose capacity is 256 bytes. Transmitting speed and rotary speed are 125K bytes per second and 200 RPM, respectively. Retaining the features of discs now marketed, the diskette unit has greatly reduced the operational procedures to facilitate operation by the general user. Three diskettes can be simultaneously connected to the terminal. (Second and third diskettes are optional.)

New hardware, to be available in the near future, include P-ROM writers and ICEs.

SOFTWARE

Several applications programs written in BASIC are recently available on diskette, including personal finance, payroll, math tutoring, backgammon and blackjack.

SORD Computer Systems Corporation is developing new products to diversify the M200 series. New applications software includes general ledger, accounts receivable, music theory, programming tutoring, inventory control and others. In addition, advanced system software such as FORTRAN IV, multi-BASIC, editors and assemblers are already in the works.

DEDICATED UNITS OF THE SORD

THE M230 LAB: In addition to standard features of the M200 series, the M230 Lab, a laboratory-oriented model, is also equipped with A/D and D/A converters. It permits the analog-digital processing.

THE M250 CCR: The M250 CCR is a computerized cash register for use in retail outlets, supermarkets, department stores, gas stations.

On the basis of cash registering functions, it features the data processing and visual response on a series of daily sales statistics. It can link to conventional electric cash registers which are in many supermarkets and chain stores. By using an optional Data Collector, data registered in each ECR can be directly transmitted to the M250. If necessary, the M250 CCR can be connected to a main computer in a head office or to a terminal unit at a stock controller.

THE M270 TSS: The M270 TSS is a terminal unit for a Time Sharing Computer System. By use of a close circuit or telephone lines, it can be connected to a remote large scale computer. Transmitting speed is 300 bps or lower. Without any modification, conventional telephone lines can be used.

Messages and data sent from a main computer can be outputted on the video monitor screen and even printed out. Furthermore, it can send data stored in the built-in minifloppy diskettes to main computer with extremely high speed. Data sent into the M270 TSS from main computer can also be stored in the diskettes.

This line of products will be marketed in Japan in the winter of 1977 and abroad in the spring of 1978. The new personal computer will be priced at under $6495.

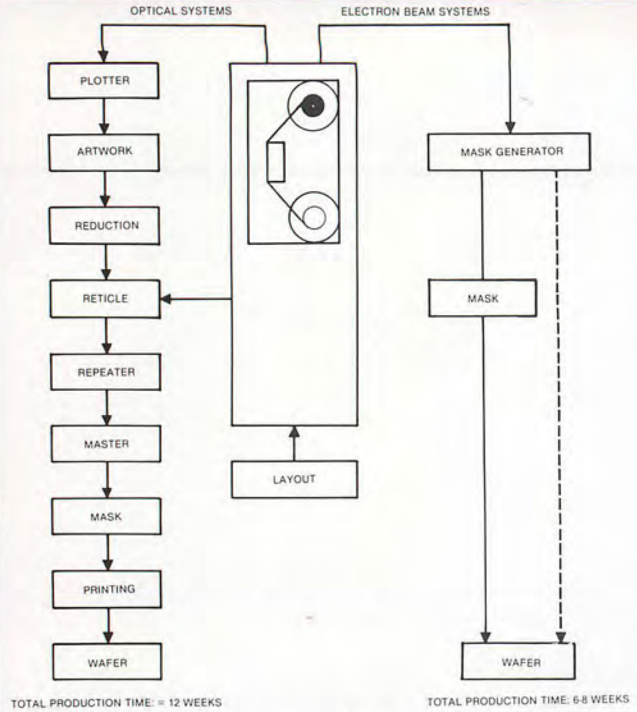


Figure 1. IC Development Cycle

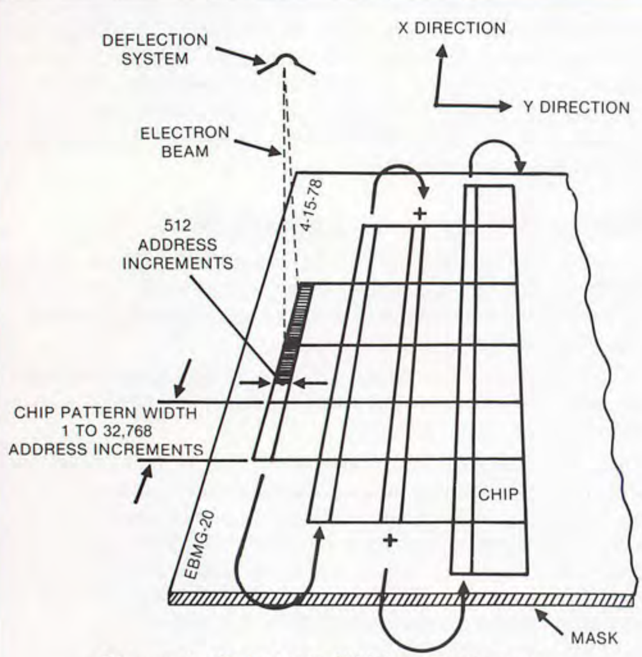


Figure 2. Substrate Writing Procedure

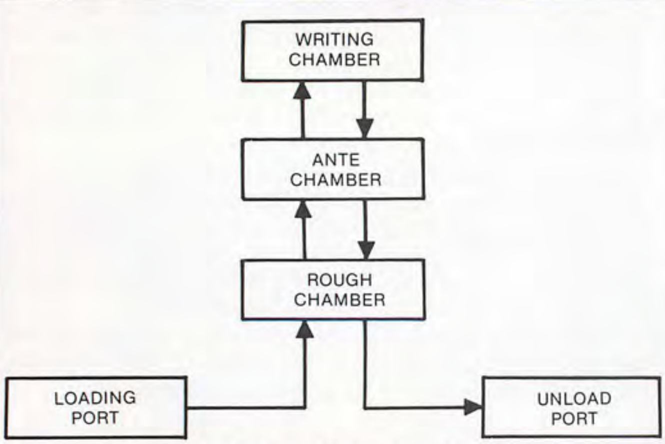


Figure 3. Automatic Cycling Process

Electron Beam Produces Nine Times

By Dr. Andrew Wittkower

An electron beam lithography printer permits users to manufacture ICs with linear details down to one micron in width. The linewidth capabilities of this system produce chips almost nine times smaller than by conventional means.

The accuracy of the system is claimed to permit denser packing and faster operation than by any previously employed lithographic process. (See block diagram, Figure 1.)

Patterns can be written on masked plates up to 6x6 inches and on wafers up to 5 inches in diameter. A 3-inch wafer mask can be produced in about thirty minutes.

DESCRIPTION OF THE SYSTEM

The system is the product of Varian Extrion Division and is designated EBMG-20. It consists of two major components, the writing system shown in Photo 1 and the control electronics. The writing equipment includes an electron gun, an electron column, a writing stage, a stage drive, interferometric devices, a loading chamber and a vacuum system. All these elements sit upon a vibration-absorbing base.

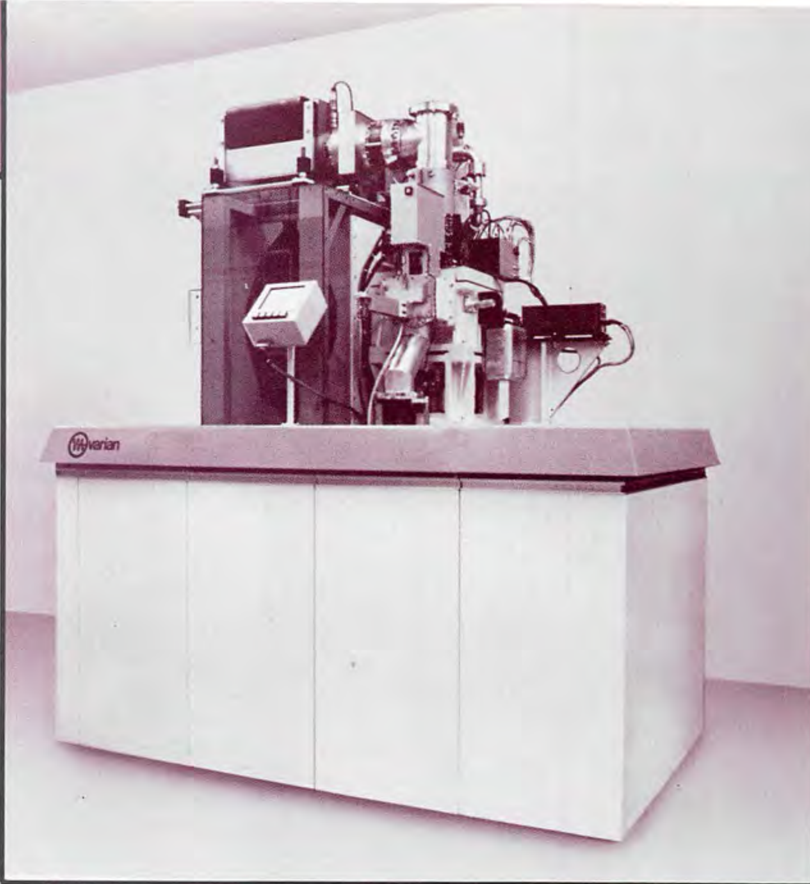
The control electronics comprise an operator console, a DG Nova 830 minicomputer with peripherals interfacing into the subsystems, a microprocessor, pattern memory, power supplies and CRT displays. The minicomputer has overall control of the writing subsystems, directing and monitoring alignment and spot size parameters. The NOVA also directs data flow from the pattern library to the beam blanking bit stream.

LITHOGRAPHIC PROCESS

Pattern generation takes place at the writing plan on the work stage under the electronic gun and the electron column (See Figure 2). The electron gun supplies a beam which is focused, scanned and blanked in the column under the direction of the control electronics. Patterns are exposed on substrates by electron-induced chemical bond changes in special resists on the substrate surfaces. Either positive or negative resist can be used to provide clear or opaque patterns. Exposure of actual pattern shapes is accomplished by electrostatic blanking of the electron beam as the beam is scanned raster fashion.

The electron column contains beam-limiting aper-

Generator ICs Smaller



tures, two electronic condenser lenses, and an objective lens as well as the beam blanking and scanning systems. In addition to the primary electron optical elements there are three sets of centering coils to permit precise alignment of the electron beam with the lens axis. Stigmator coils located ahead of the objective lens correct any astigmatisms produced in the primary optics. The degree of astigmatism is measured at the image plane and adjusted by the computer.

Beam blanking is done with electrostatic deflector plates which have response times of less than five nanoseconds. Blanking information is loaded alternately into two 512-bit buffer memories from the main pattern memory and applied to the blanking plates in synch with the raster scan. The magnetic scan system in the objective lens covers a scan field in the write mode of 280 microns by 256 microns. Response time of the deflection system is 0.5 microseconds. In the scanning electron microscope (SEM) mode, a field from 25 to 1000 microns square can be scanned to locate the fiduciary marks.

The system can expose the basic pattern for a mask for a 5" diameter silicon wafer in about 70 minutes. The minimum linewidth possible is one micron, with a tolerance of ± 0.15 micron. Registration on a substrate is ± 0.125 micron with acceptable fiduciary marks.

The writing stage is a lightweight aluminum table which accepts cassettes holding either wafers or master hard chrome mask blanks. Dynamic elements of this X-Y stage are designed to have the minimum possible mass and maximum possible rigidity. The first mechanical resonance of the stage is well above 100 Hz. Because of this design, frequency vibrations which could displace the stage under the electron beam are not able to couple to stage.

DC torque motors position the stage. These have greater resolution than conventional stepping motors. Air bearings with outer diameters machined to within 0.00025 inch of the inner diameter of their stators support the moving structure.

X and Y stage positions are measured by a dual axis laser interferometer. Zeeman splitting of the helium-neon laser line (6328 Angstroms) produces two frequencies which are double-reflected to cause a Doppler shift

of the frequency difference and provide a basic accuracy of ⅛ the laser light frequency or 0.0791 micron. Hardware logic manipulates this accuracy to 1/24 the laser frequency or a 0.026 micron-extended resolution. The Doppler shift measurement of the frequency difference has the advantage over conventional interferometry in being insensitive to the large light intensity variations common in most laser systems.

Corrections to stage position under the electron writing beam are fed to the DC motors, residual stage errors are adjusted by computer-controlled correction of the electron writing beam. By this means mechanical position errors up to ± 128 micron in X and ± 12 micron in Y position can be electronically compensated.

Stage registration is accomplished by scanning the intersection of two wires in a copper-rhodium grid mounted above a Faraday cup. The coordinates of this intersection and the grid orientation with respect to the stage axes are determined initially and stored in the computer memory. Stage position is automatically monitored while a mask is being written to assure that translations caused by thermal and electrostatic variations have not occurred. Any drifts greater than 1/16 micron automatically generate an error message on the console screen and call for operator action. During direct writing on a wafer, three fiduciary marks on the wafer are read and the pattern is brought into alignment.

The vacuum system maintains a continuous 5×10^{-7} Torr vacuum in the writing chamber. The writing vacuum equipment is separated from the loading chamber system to allow loading without interrupting writing operations. All vacuum systems, writing chamber, loading, electron column, electron gun, are isolated to permit normal maintenance without disturbing adjacent systems. All are interlocked to protect components in the event of electrical or cooling failures, and all pumpdown sequences have been automated to protect them against operator error.

PRODUCTION OF CHIPS

Mask plates or silicon wafers are handled in manually-loaded cassettes in batches of ten to supply eight hours of continuous operation in a single vacuum cycle. The mix of substrate types and sizes is handled by the substrate handling system.

Production operation centers around the operator's console and the loading assembly. After beam parameters have been set up, only the keyboard controls, CRT and error display are used. The computer is instructed, cassettes are loaded into the vacuum load lock and the automatic cycle is initiated. All system operations are protected against operator error, vacuum, air or power failure.

During the automatic cycle a magazine of 4 to 10 cassettes is placed into a loading port. These are elevated to the roughing chamber which is then pumped down to a rough vacuum. The magazine is then shuttled into an ante chamber which is maintained at high vacuum. The individual cassettes are then moved into the writing chamber, processed and returned to the ante chamber, where the resist is allowed to develop while another cassette is being processed in the same manner. This allows continuous processing. When the entire magazine has been processed, it is moved back into the roughing chamber, then to the unload port. (See Figure 3, block diagram).

CONTROL ELECTRONICS

The Nova 830 minicomputer performs correct sequencing and control of all the various machine functions and translates operator commands from the control console and loading control assembly. Manual control for maintenance and service is obtained through special software programs.

Peripherals for the computer include a disc for pattern storage, a 9-track tape system for pattern loading and dumping, an electrostatic pattern plotter, card or punched tape reader, and a printer terminal for pattern maintenance. An 8-bit, 5 MHz microprocessor converts pattern information into a beam blanking bit stream and creates a graphic art display including SEM reproductions.

Pattern writing is accomplished by taking data stored as trapezoids from the disc storage system, and converting to an "on-off bit map." The data are applied to the beam blanking system and synchronized with the scanning system.

Pattern data are stored as strips 512 address locations wide by 16,384 address locations long containing an arbitrary number of arbitrary shaped trapezoids. Under control of the computer, the microprocessor converts the trapezoid data to 1's and 0's completely filling the strip. The data strip can be truncated in length, to adjust it to the length of the X dimension. The binary data are stored in a 512K-word core memory which can be read sequentially from either end, and block transferred to an intermediate buffer memory. The core memory transfers occur one-half at a time, to permit loading one-half from pattern storage while the other half is sent serially to the beam blanking system.

A 16-bit shift register, clocked at 20 MHz, supplies the final buffer memory and converts the 16-bit words to final beam blanking information.

The data conversion process, from trapezoids to single bit information, takes a few seconds maximum per strip and is done while the stage re-positions itself to start the next strip. Writing is done in serpentine mode from top to bottom, and left to right, then right to left, over the surface of the substrate.

As a strip is written, address registers in the stage control system are hardware incremented in the X direction to provide successive raster scan lines with correct spacing. The actual value of scan is adjusted according to information received from the fiduciary mark detection subsystem. The stage control system attempts to maintain correct location for each "line" in the Y direction. Any deviation from the correct position is noted by the laser interferometer. This information is used by the computer both to adjust the stage and to correct the electron beam deflection to maintain the writing tolerances. (See Figure 2)

THE COMPUTER IS THE LIBRARIAN

Original IC mask pattern data are handled in two different forms. Feature information is inputted on magnetic tape through the 9-track tape system in either optical pattern generator form or by a special and more efficient electron beam lithography format. Alphanumeric and array information required to assemble patterns in to complete mask levels is inputted through the punch card or paper tape reader in a special format.

A complete software package is supplied with the system for production, pattern design and management, system set up, and for hardware maintenance.

CONCLUSION

Although the system price EBMG-20 is very high, approximately \$1½ million, the timesaving advantages of this equipment might prove a good investment to the chip manufacturer ultimately leading to a price cut of semiconductor products on the retail level.

FOOTNOTE

The documentation for this article was supplied by Varian, Extrion Division, Palo Alto, California.

The WRT System: From Field Notes to Programmable Data

By Dr. S.J. Poray-Tucholski

Under field conditions statistical notations are normally entered into pocketsize notebooks later to be transcribed onto formsheets. This method lends itself to timewaste and the possibility of errors introduced into the data.

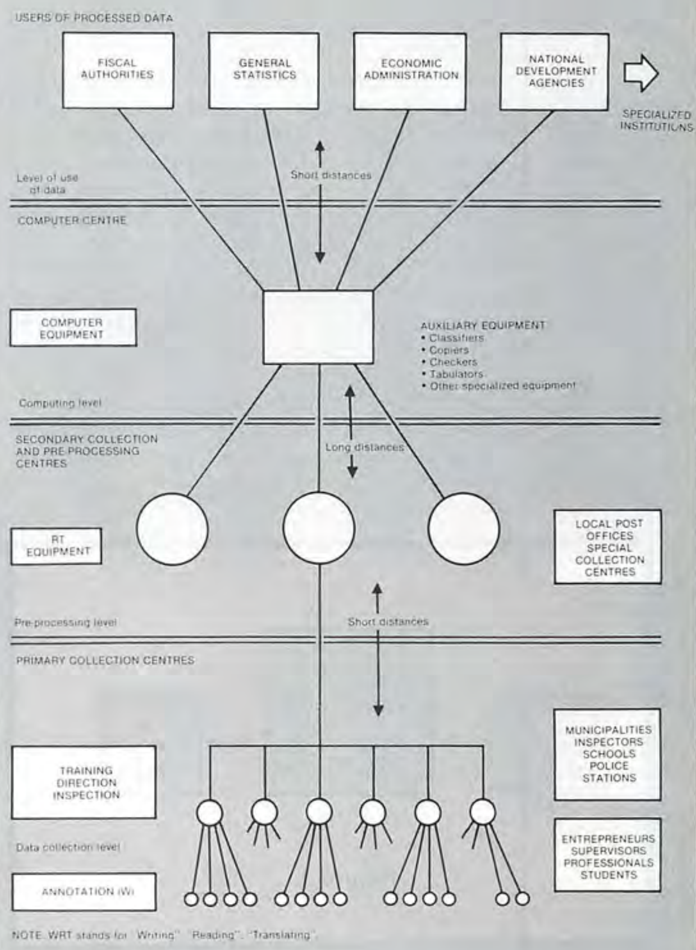
Dr. Zygmunt Slawinski of Chile has devised a method by which primary notes may be entered into computer readable tape. The annotation is made with an ordinary pencil with specially-treated accordion-folded paper of notebook width preprinted with ten columns of computer-readable signs such as squares, circles, ovals, rectangles or other simple shapes. Alongside the columns the tape has a free space for additional documentation to be later entered into the program. Data are entered into the tape by covering the specific surface symbolizing the datum. The tape is later decoded by an optical reader and the information transferred onto magentic tape or disc for storage.

While the principle and application of this method of data entry are well-known to any U.S. voter, its application in non-industrial countries is virtually untried. In those areas census-taking, ballot tallies and disaster information suffer serious delays because of hand processing.

With the aid of the easily portable and inexpensive microcomputer, data collection centers can be set up in any location accessible to electricity and telephone lines. Using the microcomputer as a terminal, the data can be transmitted via telephone lines to centralized banks for accessing by the user agencies.

Much of the scheme is still on the drawing board. The pilot study for Chile is to be carried out under the auspices of the Latin-American Institute of Economic and Social Planning (ILPES).

WRT DATA COLLECTION AND PROCESSING SYSTEM



Flowchart of the Complete System

Editor's Note

The above news report was summarized from Dr. Slawinski's project proposal *Notes for the Technical and Economical Feasibility Study of the WRT System for Collecting and Processing Numerical Data* first submitted to the United Nations in 1968. Since then Dr. Slawinski has been at work on the project in Santiago, Chile. Though the project suffered a slow start, the decreasing cost of electronic equipment in the past two years will accelerate implementation of this plan.

Should any of you want to contact the inventor, he can be reached by writing to: Dr. Zygmunt Slawinski, 2626 41st N.W., Washington, D.C. 20007.

An Introduction to the Computer

By Roger Edelson, Hardware Editor

The intent of this series of tutorial articles is to provide familiarity with that strange beast, the computer! Depending on how you may use a computer, your view of what a computer is will be different. Those of you who use a company computational facility by means of job submittal cards do not really see the computer. All that is seen is a window into which a job is submitted, and from which the result is obtained. At the other end of the spectrum is the circuit designer whose responsibility is one small function block of the computer and never sees the computer as a problem solving machine at all.

Obviously, most of us fall somewhere in between and would like to understand the computer as manipulative machine which may be applied to various problems. It is to this end, or basically the view of the home or small business computerist, that these articles are addressed.

To most of us the computer will be a varigated assemblage of circuit boards, chassis, and input/output devices. In order to use these assemblies we will have to take a look at what they are and how they interact.

Let's start by taking a look at the general architecture of a stored program computer — which is the real name of the beast with which we are dealing.

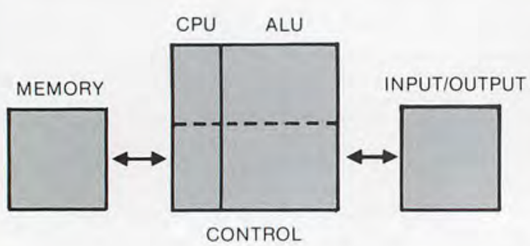


Figure 1.

Pretty simple — huh! This diagram does not actually represent the typical microcomputer with which we shall be dealing because it does not show the typical BUS structure. Actually, with the exception of the input/output device all of the major elements (CPU and Memory) could be placed on one chip. In some of the newer control-oriented machines they are. In the type of machines with which we are usually involved this is not practical because of the limitations to expansion. Prior to the advent of microcomputers and solid-state memories a much larger piece of real estate was required. As a historical note, the UNIVAC 1 which was built out of vacuum tubes in 1950 and filled a room, had less computing powers than most of today's microcomputers.

Let's take a somewhat more detailed look at the various pieces which make up the microcomputer block diagram. In this article the look will be basically an overview with subsequent articles examining each section in considerable detail. We shall begin the overview with the CPU — the Central Programming Unit.

The Stack and Stack Pointer are used to modify the memory...

The CPU is the essential heart of the microcomputer system. With the advent of the single-chip CPU (in essence a whole computer on a slice of silicon) home computing came of age. The basic function of the CPU is to perform calculations, memory handling, and various logical functions. The CPU of a typical microcomputer generally contains one or more accumulators which are used to store data retrieved from memory. It is in the accumulator register that the data fetched from memory are operated upon. The accumulator also serves as a holding point prior to returning data to memory. It might be possible to obtain a word from memory, operate on it, and return it directly to memory in an instruction cycle but the additional significant amount of logic required to implement this operation would be substantial. The CPU will also have a register which stores the address of the data in the memory. The size of this register will determine the extent of the memory that can be directly accessed by the CPU. (Other schemes can be used to extend this address field.) In most of the CPU's we shall be concerned with, the memory size will be 65,356 words — equivalent to a 16-bit address field. In many CPU's this register is also called the Data Counter.

The CPU must also have a place for storing the instruction to be implemented. This is called the Instruction Register. The CPU will always interpret the contents of the instruction register as an instruction. On the other hand the CPU does not care what the bits of data stored in the accumulator represent. They can be either instructions or data.

Another register is always required, the Program Counter. The Program Counter is assumed to always refer to a program memory word and specifies the next location from which the instruction will be taken.

Most CPU's also contain one or more Index Registers which may be used to store data or a memory address for

Tutorial — Part I

World of Home Computers

use in Indexed Mode memory addressing. Again most modern CPU's feature a register called the Stack Pointer. The Stack Pointer indicates the next available location in an external transient storage area, usually the lower address of the system memory. The Stack and Stack Pointer are used to modify the memory addressing technique in a manner called implied memory addressing.

Almost all of the CPU chips which we shall cover in a subsequent article are bus-oriented devices. By this we mean that the data (usually 8-bits) are transferred both into and out of the CPU by one set of eight lines through the use of bidirectional drivers. The same is true for the Address Bus — usually consisting of sixteen lines as necessary to handle the 65K memory space. Many of the CPU chips also have a control BUS structure for some of the specialized control functions necessary. Figure 2 illustrates the implementation of the typical microcomputer BUS structure.

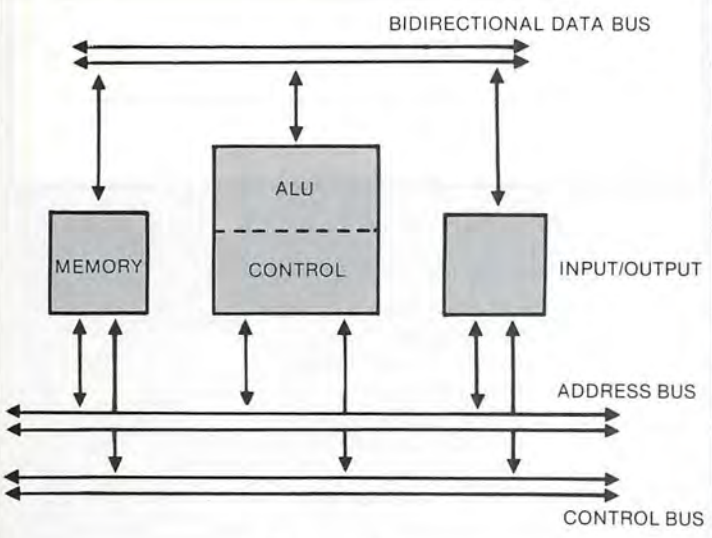


Figure 2.

The second block of Figure 1 represents the external memory function. The Memory provides a storage facility for data words and program/instruction commands. The size of the memory space directly addressable by the CPU is set by the number of bits provided in the Memory Address Bus. Most of the CPU's we use have a 65K memory address capability. This can be expanded by the use of additional mass storage devices which access the CPU through the input/output function. The memories available range in both function and technology used to implement the particular function. RAM stands for Random Access Memory which usually is the working memory of the computer. This style of memory may be either written into or read from under program control. Depending on the technology used the information may be lost upon reading (i.e., magnetic cores) or retained in storage after a read (semiconductor memories are an example). Again, depending on the technology used the memory may be volatile (information is lost when power is removed), or non-volatile. Semiconductor memories with some exceptions are an example of a volatile memory and magnetic cores are non-volatile.

ROM refers to Read Only Memory. This type of memory is non-volatile but can only be read from. ROM is preprogrammed during the factory manufacture by the selection of an appropriate mask. Some memories are programmable in the field — PROMs, Programmable Read Only Memory. Some of the technologies used to implement PROMs allows one write operation only — the device then becomes a ROM. There are technologies which allow data stored in a PROM to be erased and new information written. In most of the available devices the erasure is done by ultraviolet light. There are other devices that can be erased electrically. Most of these devices are limited to single to thousands of bits. On the high end of the spectrum are the mass storage memories which are usually used as peripheral devices. In this category we will find tape recorders, paper tape, discs, both rigid and floppy, bubble memories, etc. In a later article all of the various competing memory technologies will be examined.

The Input/Output function provides the communication between the CPU and the outside world of peripheral devices, i.e. readers, punches, displays, keyboards, etc. Without some I/O function there would be no way to access the computer nor to obtain the results of a computation. The I/O function is also used to provide access to additional mass memory devices such as floppy discs, and magnetic tape recorders. Most of the I/O peripheral devices are not hung directly onto the bus structure because they have data rate requirements much slower than the CPU and its associated memories. Instead one of the communications interface adapter chips is used. Also since many of the peripheral devices interchange data serially (a bit at a time) rather than parallel (8 or 16 bits at a time) some provision must be provided to convert the parallel data transfer from the CPU to a serial string at the rate required by the peripheral device.

Tying all of these functions together is the Program. It is the program which tells the computer what tasks to perform and details how each of the various functions are to be used.

In subsequent articles we will look in detail at each of the above functions. Next month we will look at the various CPU's available and what functions each can provide.

Build A Light Pen

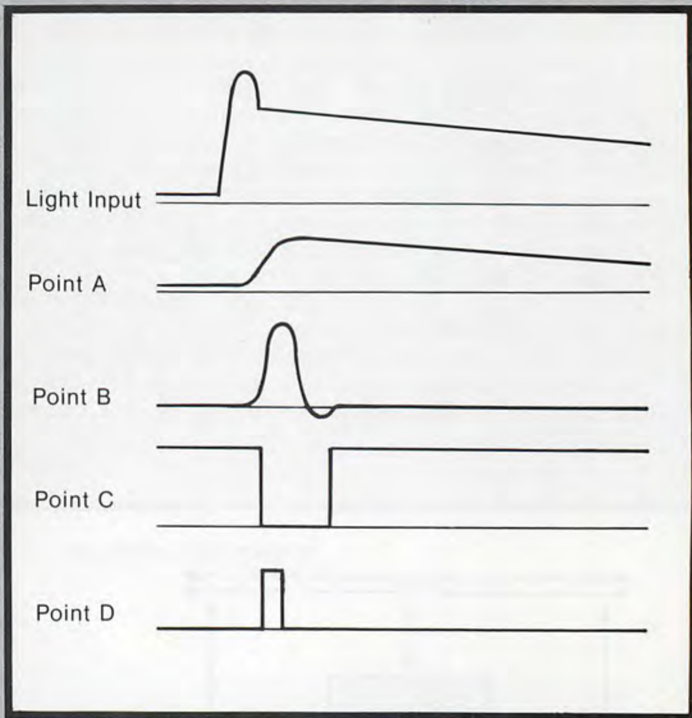
By Scott Fluhrer

There are many interesting peripherals you can connect to your computer, including analog inputs/outputs, TV cameras, joysticks, speech synthesizers, AC power controls, music synthesizers, plotters, printers, and frequency counters. One might think they have nothing in common, but they do — cost. They all are fairly expensive. On the other hand, how would you like to build a light pen for approximately thirty dollars? No, this is not a construction article, and don't believe I am the ultimate expert on this subject, I'm not. I have not built this circuit because I don't have a computer. However, feel free to use, discard, or elaborate on any of the ideas here.

This light pen operates on the principle that when the electron beam passes over the phosphorous in a TV tube, the phosphorous quickly lights up. (See Figure 1.) Although most phototransistors couldn't follow the light input exactly, a good one should give an approximation, as in Figure 1-Point A. The signal then goes through C1 to screen out the effects of ambient light. The comparator IC 1 then changes this analog pulse into a digital one. IC A-B then shapes the digital pulse. At the same time, the video circuitry is reading memory to decide which character to output. ICs 4-7 store the address of the memory being accessed and therefore the address of the character to which it is pointing. The computer can read the latches via two input ports.

A critical part is the phototransistor. To be reliable, it must have a rise time of at most one microsecond, and preferably less. R1 has been calculated assuming an average (between maximum and minimum) phototransistor resistance of 50K. If yours varies drastically from this, you should alter R1, R2 and C1. SW1 is the on-off switch. I suggest you put it on the pen itself in a convenient position and it could be a very nice feature. To adjust R3 correctly, you take an oscilloscope and measure the peak voltage at Point B. The armature of R3 should be a bit less than that. The input lines connect directly to the video circuitry, to the character counter not the curser. One clue: the counter is usually connected to a high speed clock.

A light pen is very versatile. Can you imagine what kind of text editor you could have with it. Point at a word, press a button and the word disappears, or it moves, or it underscores a word. A light pen would make computer art a snap. You could draw a picture and have the computer distort it, perhaps. But, most importantly, a light pen would add a new dimension to computer games.



Parts List for Figure 2 Schematic

C1,C2 — 100 pF. Cap.
IC1 — Voltage Comparator
IC2 — 7402 Quad NOR Gate
IC3 — 7404 HEX Inverter
IC4-7 — 7475 Quad Latches
IC8-11 — 74125 Quad Tristate Buffer
IC12 — 7430 8-Input NAND Gate
PT1 — Phototransistor
R1 — 50K Resistor
R2 — 500 K Resistor
R3 — Pot (most any value)
R4 — 500 Ohm Resistor
SW1 — NC Miniature Pushbutton Switch

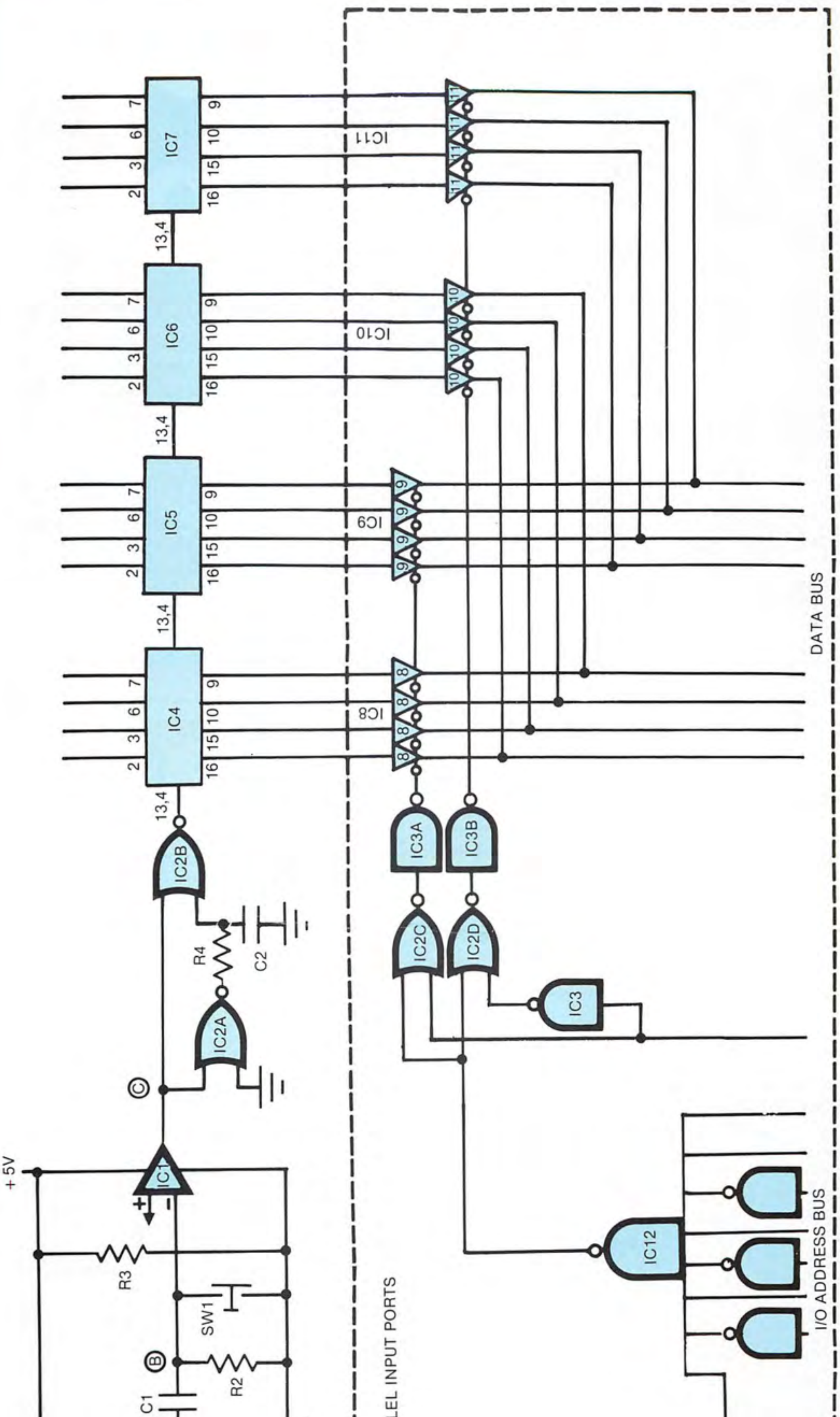


Figure 2. This is the schematic for the proposed light pen. Note that the lower half is just two parallel input ports.

A Keyboard/ Peripheral

By Larry Grim

INTRODUCTION

Interfacing pushbuttons and 7-segment displays to a microprocessor is the least expensive way to get human input into the microprocessor and visually readable output from the microprocessor. Several computer kits have used this technique instead of a front panel or a terminal to keep costs down. The ROM monitor supplied with the kit constantly scans the pushbuttons looking for closures and takes appropriate action whenever a button is pushed. Likewise, whatever information is to be displayed will be written out to the LEDs.

In spite of the low cost, this means of human input and output seems to be relegated only to high-volume, low-cost applications. Hobbyists and even design professionals seem to think in terms of printing terminals and CRT displays instead of a custom-designed keyboard and display for a specific application.

There are some very good reasons for this, however. While one can purchase a complete terminal which easily interfaces to a microprocessor, as far as I know, standard interfaces just don't exist for either pushbuttons or 7-segment displays. Also, while reading keyboard input from a terminal device is absurdly easy, scanning a switch matrix is not and to do it requires some software. Finally, the terminal interface can interrupt whenever it receives a character, allowing the CPU, while waiting for a character, to do whatever it desires (or maybe what it was programmed to do). The scanning of the pushbutton matrix is entirely done in software and ties up the CPU.

To demonstrate that keyboard-display interfaces can be simply and easily designed and constructed, some basic hardware information about keyboards and displays and the connection to microprocessors must be presented. Then, a new part, the 8279, will be examined that is expressly for interfacing keyboards and displays to microprocessors. This part will alleviate the inherent problems that are uncovered in the basic information describing interfacing to the microprocessor.

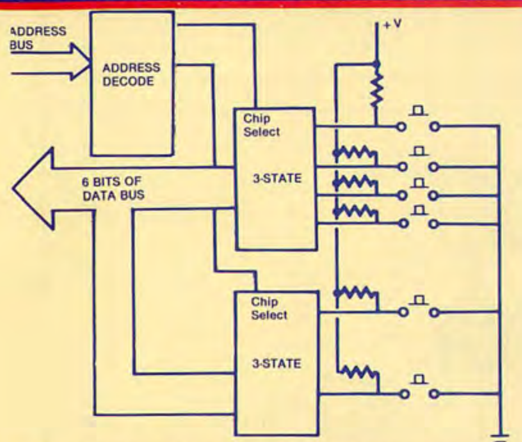


Figure 1. Twelve pushbuttons directly connected.

THE BASIC PUSHBUTTON INTERFACE

For a small number of switches, it is feasible to connect each pushbutton directly to an input port. In Figure 1 the pushbuttons are connected to HEX 3-state buffers, the CPU getting the values of the six switches on a single read of the 3-state buffer. Since many HEX buffer chips are enabled in two groups, one of four and one of two gates, the data bus could easily be used as eight bits wide instead of six by using two gates in a separate IC with the same address decode as is used by the first six. With an octal address decoder, up to 64 pushbuttons could be read in, requiring eleven HEX buffers and much data bus wiring. One advantage to this scheme is that all the buttons can be depressed simultaneously and the CPU can sense each and every one.

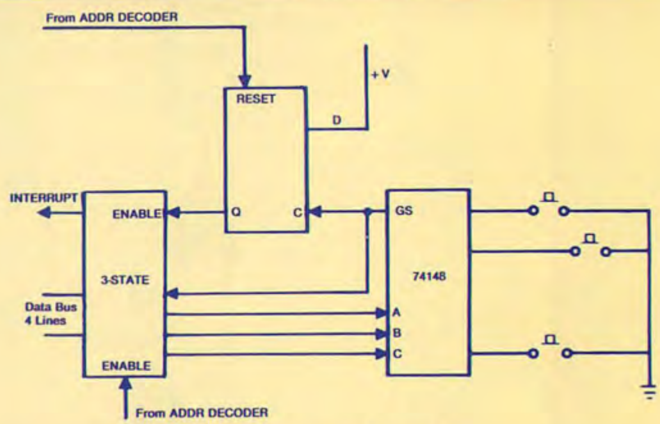


Figure 2. Encoded inputs.

A different way, that uses fewer inputs into the microprocessor, is to encode the pushbuttons. The input that is given to the microprocessor is a coded representation of which of the buttons is depressed. A priority encoder, as in Figure 2, works well here. There are two improvements in this circuit over Figure 1. One is that the switches will be encoded on an input, that is, the number of the switch will be input directly. The other is that this circuit has an interrupt line and will wake up the CPU whenever a button is pushed. This frees the CPU from having to scan for key pushes.

The GS line from the 74148 goes low whenever there is a key pushed and stays low for as long as it is pushed. By hooking GS to the interrupt line of the processor, an interrupt is generated when a key is pushed. (It would actually interrupt on the opening of the switch most of the time because of contact bounce.) The circuit has one problem. The 74148 output for switch number "0" and for no switch pushed is the same (all high). To distinguish between these two outputs, the GS line is included on the data bus. This also allows the switches to be polled. An alternate scheme is not to use switch position "0" so that an all high output uniquely means

Display Interface on a Chip

no switches pushed. The disadvantage of the priority encoders is that only one switch can be sensed at one time, even if two are pushed. Also, if two are pushed and one is released, no interrupt is generated because GS stays low. To handle more than eight switches the encoders can be easily cascaded by using multiple chips (up to 64 switches). There is also a 74147 which handles up to ten keys on one chip instead of eight, but it doesn't have a GS signal and can't be cascaded.

The technique normally used with a larger number of switches is to form them into a matrix and scan them one row at a time looking for a closed switch. In Figure 3a a 4x3 keyboard is scanned, using only four digital outputs and three inputs. One of the digital outs is set low and all other high by the CPU. By reading the inputs, the exact row and column of the closed pushbutton is known. Since only one line can be low at a time, the CPU doesn't even have to do the encoding. Figure 3b shows a 2-to-4 line decoder which takes the CPU binary output and converts it so that one line at a time goes low. For larger matrices, 3-to-8 or 4-to-16 line decoders can be used. The disadvantage here is that the CPU must scan for key pushes and the scanning program will be more complex.

The terms *switch, pushbutton* and *key* have been used interchangeably because there is no reason that the contact closure be a momentary type; it can be any kind of on-off switch. It doesn't even have to be a contact; a transistor driven by other electronics (real world inputs) could just as easily serve. A single matrix could have all three types mixed and satisfy all the input needs of a small system.

There are several ICs available to handle scanned keyboards. Available from National Semiconductor are a 90-key keyboard encoder which contains a ROM with the desired output for each key (MM5740), a 78-key keyboard encoder also ROM based (MM5745), a 16-key encoder for a 4x4 matrix (MM74C922), and a 20-key encoder for a 4x5 matrix (MM74C923).

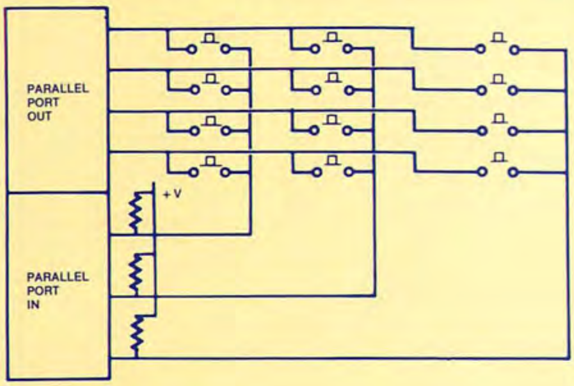


Figure 3a. Basic decoded scanning technique.

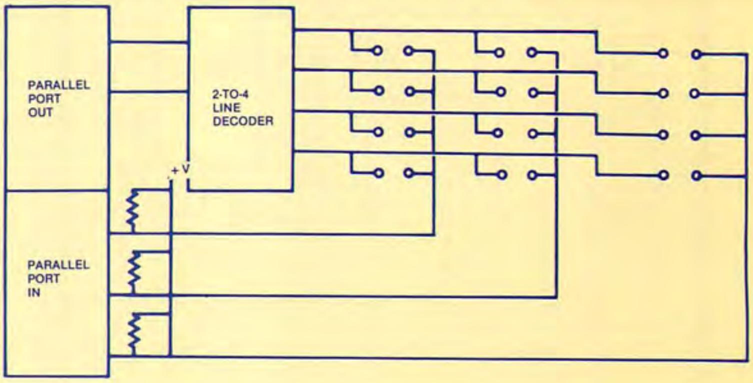


Figure 3b. Encoded scanning technique.

One final note on mechanical closures is in order. When the contacts come together or separate, they actually make and break the connection many times because they bounce and at the microscopic level the surface of the contact is not flat. The time it takes a contact to settle is typically milliseconds and during this bouncing a microprocessor would be able to detect multiple closures. There are many techniques to handle this in hardware to get a nice clean, sharp level change (for example, the Motorola MC14490 HEX Contact Bounce Eliminator). However, the CPU itself can do the debouncing in software and eliminate the need for the hardware. When a closed contact is found, the CPU delays ten or more milliseconds and then rereads the switch. If no change has occurred from the first read, the new switch position is assumed. If they don't agree, ignore the key this scan because eventually it will settle down.

THE DISPLAY INTERFACE

There are many kinds of displays that can be used (LED, LCD, projection displays, plasma, gas discharge, fluorescent, incandescent, etc.). Of all these, LEDs will be found in most small systems because they will run on the 5- or 12-V power supplies encountered in logic circuits; they are readily available from many sources; they are cheap, and they come in the widest variety of sizes and packaging. Therefore, all the circuits will assume LEDs are to be the display medium, although other types can be used in many cases.

By far the most common LED display is the 7-segment numeric display. The seven segments are arranged as shown in Figure 4 and labeled a, b, c, d, e, f and g. They are either wired as common anode, where all the power supply sides of the seven segments are connected together and the ground side is switched to turn on or off an individual segment, or in common cathode where all the grounds are connected together.

The 7-segment displays are very versatile as can be

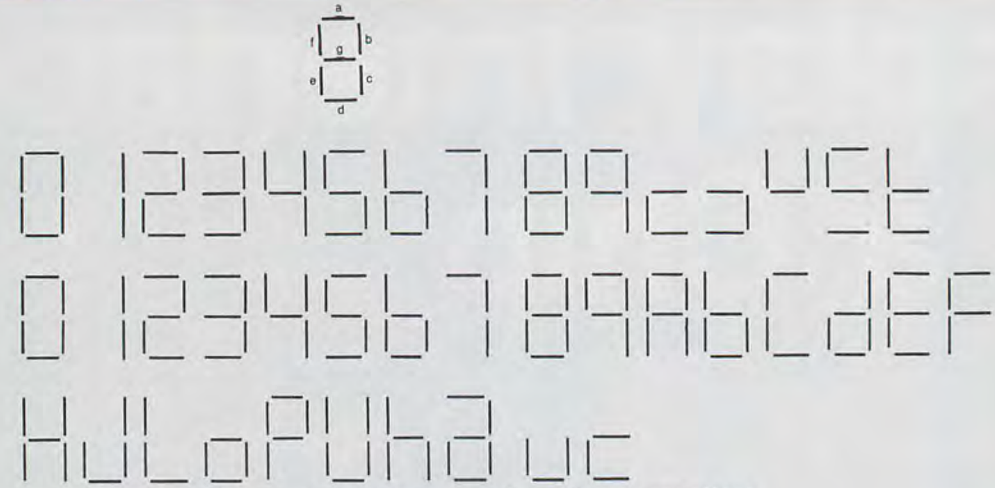


Figure 4. Seven-segment numbers and letters.

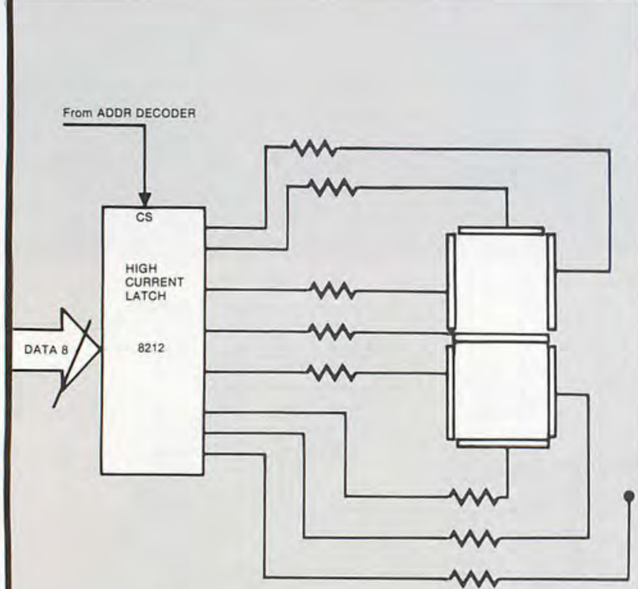


Figure 5a. Direct drive LED.

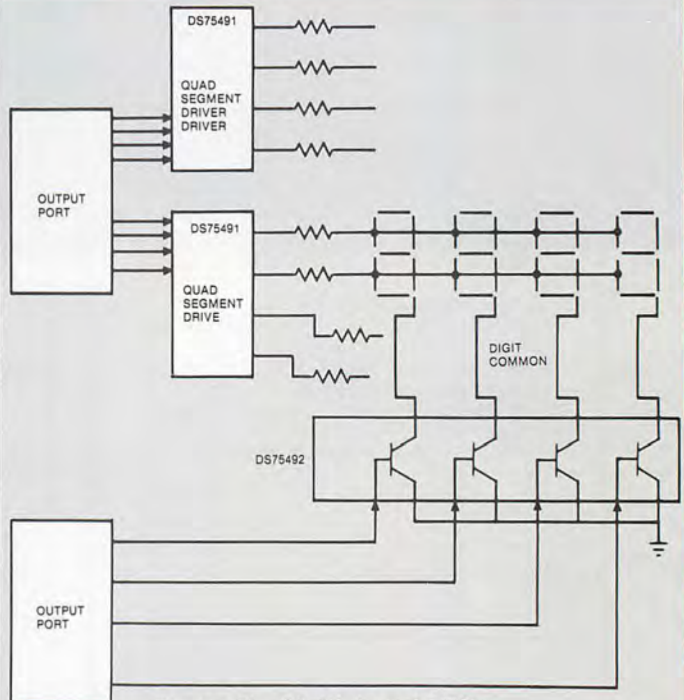


Figure 5b. Multiplexed LEDs.

seen in Figure 4. Both numeric and some alpha characters can be displayed. The first row shows the standard output of a BCD to 7-segment decoder. The second row shows the choice that would be suitable for hexadecimal character display. However, confusion will be caused by the six and the lower case "b" and "B's" will be read as "6's". Hewlett Packard, in the newly announced Signature Analyser has changed the displayed characters to 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, C, F, H, P and U so that there won't be any confusion and 7-segment displays can be used. This output however, is not used as a HEX number, but is a string of symbols that must be matched against a similar string. The last row in Figure 4 shows some more upper and lower case letters that can be formed.

The simplest way to connect the displays to the microprocessor is to have one output to control each of the segments in each and every display digit. As in Figure 5a, by using a 8212 8-bit latch and a common anode display, the latch can sink current in the low state (up to 15 mA) and turn a segment on. The resistors are necessary to limit the current through each segment. Most displays have either a right hand or left hand decimal point and it is hooked up the same way. The disadvantages of this scheme is the high cost of the drive circuit and resistors, and the large amount of wiring for the data bus and the resistors.

Another way to connect the digits is to multiplex them. Each digit is turned on for only a fraction of the time but is much brighter when it is on. For example, in a 4-digit display, each digit turned on 25% of the time at 40 mA of current through segments that are lit would be approximately equivalent to a direct drive current of 10 mA. As long as the scan frequency is high enough to produce flicker-free viewing, the multiplexed display will have apparent intensity of its average current. For flicker-free viewing, the displays must be multiplexed at least 50 Hz with 90-100 Hz being a commonly used rate.

There are several disadvantages to multiplexing. First, the current requirements go up considerably and output ports can no longer be used to switch the current. Quad segment drivers (DS75491) are shown in Figure 5b which are capable of switching up to 50 mA in each segment. In addition, each digit must be turned on and off which is accomplished by interrupting the ground line of the common cathode displays. The cur-

rent through the ground is the sum of all the segments and the decimal point, and can be eight times the individual segment current. In Figure 5b, a DS5492 switches the digits. It can sink up to 250 mA of current. A second problem is the amount of CPU time that would be required to multiplex. To multiplex four digits at 100 Hz would require an update every 2.5 ms. There is considerable risk too because if an errant program locks out the interrupts (assuming the display multiplexing is done on a clock interrupt) surely a burned LED would result because the multiplexing would stop. The value of the average current would become the peak current in the digit that the display stuck on. In an 8-digit display, the average current would jump eight times and destroy the digit. Incidentally, if the CPU were to do multiplexing, it should turn off all the digits, change the segment latch to the next digit's value and then light the next digit. This prevents the possible false display of the new segments in the last digit.

So far, the CPU has had to do the translation of the output letters and numbers to 7-segment codes. If only numbers are being displayed, then the translation can be done in hardware and relieve the CPU of this burden. The MM74C48 shown in Figure 5c has a source capability of 50 mA and can be used with 7-segment displays directly through the properly-sized current limiting resistor. The MM74C48 could directly replace both DS75491's in Figure 5b. The MM74C48 is roughly equivalent to the 7447 with slightly higher current capabili-

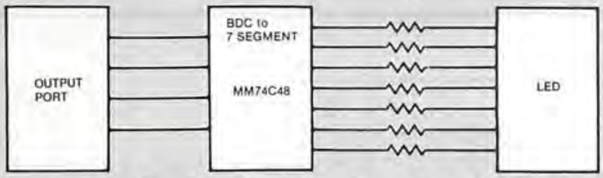


Figure 5c. Direct drive through a BCD to 7-segment decoder.

ties. The MM74C48 sources 50 mA (use with common cathode) while the 7447 sinks 40 mA (use with common anode). The first row in FIgure 4 shows the output patterns generated for binary inputs 0-15. A blank is generated from a 15 input.

The advantages of the BCD 7-segment ICs is that fewer parts are required, less wiring, less CPU overhead doing conversion and less memory to hold the conversion routines. The disadvantage is that the only non-numeric symbols available are the five offered by the conversion of binary codes 10, 11, 12, 13 and 14. Also, the decimal point is not supported on the BCD 7-segment converter and may require separate electronics to turn it on. The decimal points are frequently direct drive even when the digits are multiplexed.

THE 8279 KEYBOARD/DISPLAY INTERFACE

I recently had the opportunity to work with a new integrated circuit from Intel, the 8279 Programmable Keyboard/Display Interface, which considerably erases the burden of a keyboard/display interface for both the designer and the CPU. This part may actually make the cost of a keyboard/display even lower because of the number of components replaced. The Intel list price of the 8279 is $14.00 each in single quantity (February 7, 1977 Intel price list), not much more than your favorite serial interface chip.

The purpose of the information presented here is to give an overview of the capabilities of the chip and to illustrate how the features of the 8279 can be used. It is not the purpose to either duplicate or replace the spec sheet. It is expected that anyone actually wishing to use the 8279 will get a spec sheet to fill in the details that I have omitted here.

The 8279 is in a 40-pin package and uses a single + 5V ± 10% power supply. The ± 10% specification is unique because most of the rest of the 5V Intel parts are ± 5%. The control signals to and from the 8279 are the same as other Intel peripherals. They are shown in Table 1. To read from the 8279, $\overline{RD}$ and $\overline{CS}$ are low. To write, $\overline{WR}$ and $\overline{CS}$ are low. There is a data port and a control port, the selection of these two controlled by C/D. The C/D pin is normally connected to address line A0 so that the control port will be one location higher than the data port. There is a CLK signal which is normally connected to the CPU clock and a RESET signal. Finally, there is an Interrupt line by which the 8279 can interrupt the CPU whenever the state of the keys change.

Internally, the 8279 contains all the circuitry necessary to both read a keyboard matrix and to drive a multiplexed display. The matrix can be as large as 16x8 and as many as 32 digits can be driven. There is an

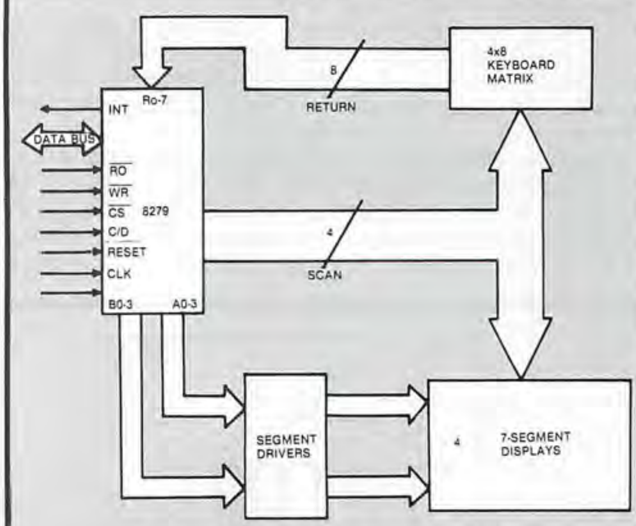


Figure 6. Decoded use of 8279.

CONTROL SIGNALS INTO 8279	CONTROL SIGNAL FROM 8279
$\overline{RD}$ — Read	INT — Interrupt
$\overline{WR}$ — Write	
$\overline{CS}$ — Chip Select	
C/D — Command/Data	
RESET — Reset	
CLK — Clock	
DB0-DB7 — Data Bus	

Table 1. CPU to 8279 Signals.

8x8-bit RAM used with the keyboard and a 16x8-bit RAM used with the display.

Figure 6 shows how the 8279 can be used in its most elementary form. The 8279 internally drives the four scan lines low one at a time in sequence (called decoded mode). The scan lines are connected to the four digit drivers and to the rows of the keyboard. The contents of the first four positions of the display RAM are multi-

plexed out through A0-A3 and B0-B3. The CPU gets the proper 7-segment codes into the first four display RAM locations and the 8279 takes care of the multiplexing. At the same time, the 8279 is reading the return lines R0-R7 from the keyboard looking for changes. The 8279 has internal pullups on the return lines.

The keyboard has been enlarged to 8x8 and the displays increased to 16 in Figure 7 by using the 8279 with encoded scanning. The 8279 uses the scan lines as a binary up counter output and will count either 0-7 or 0-15, depending on the number of displays for which it was programmed. In either case, the keyboard will use only the low three scan lines and may have only eight rows. If the display uses eight or fewer digits, then both the display and the keyboard can share the same 3 to 8 decoder. Some digit drives need active high which would require an active high 3 to 8 decoder or inverters on the one used by the keyboard. The keyboard must be active low.

THE 8279 FIFO/SENSOR RAM

The 8279 receives keyboard information from the return line inputs and two other lines, the SHIFT line and the CNTL/STB line. How it uses the information received on these ten lines depends on the input mode that the CPU has chosen. There are three modes.

The most elementary is the Strobed Input mode. On the rising edge of the CNTL/STB line, the values on the eight return lines are entered into the FIFO. The FIFO has eight positions and on the next signal from CNTL/STB, the next entry will be made into the next available position of the FIFO. When anything is in the FIFO, the Interrupt line will be set. As the CPU reads bytes from the FIFO, that entry is deleted and the other entries are moved to the front of the FIFO. The 8279 will also give the status of the FIFO, that is, the number of bytes in the FIFO, whether it has been overrun (tried to put a byte into a full FIFO) or underrun (tried to read from an empty FIFO), and a bit to indicate it is full. The SHIFT line is ignored.

The second is the Scan Sensor Matrix mode. In this mode, the 8x8 RAM is used as a Sensor RAM instead of a FIFO. As the 8279 drives the scan lines it will enter the information from the return lines into the corresponding position of the Sensor RAM. The Sensor RAM will be continually updated whether or not the CPU reads it. The decoded 8279 will use only the first four while when encoded, it will use all eight RAM positions. To read the Sensor RAM, the CPU issues a Read FIFO/Sensor RAM command and then reads the data port. The command contains the position in the RAM to be read. The command also allows auto-incrementing to be specified, which means that consecutive data read with no intervening command will cause consecutive Sensor RAM positions to be read instead of the same position twice. The interrupt will be triggered on any change of an input, either on or off. Once a change is detected, further writing into the Sensor RAM will be inhibited until the CPU gives an End Interrupt Command.

The third is the Scan Keyboard mode. In this mode, the RAM acts like a FIFO as in the Strobed Input mode.

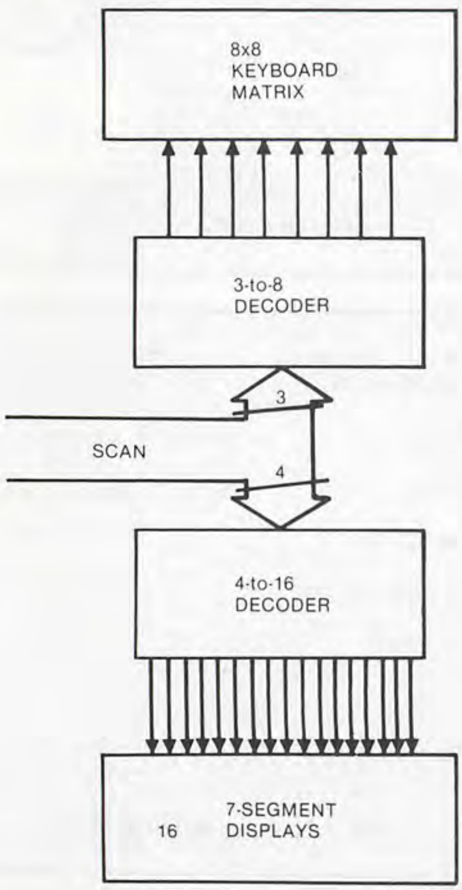


Figure 7. Encoded use of 8279.

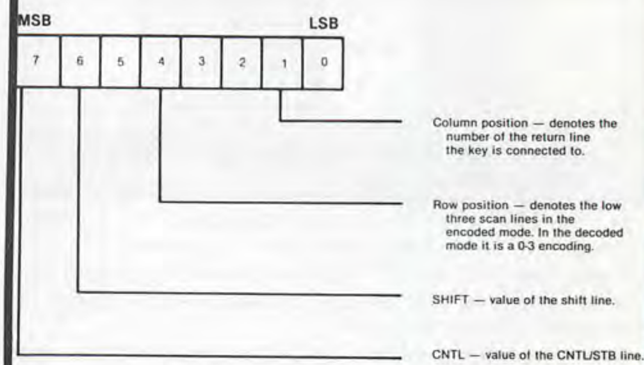


Figure 8. FIFO entry in Scan Keyboard mode.

However, the data put into the FIFO are entirely different. Figure 8 shows the format. The SHIFT and CNTL/STB inputs would have a single key hooked to them, not part of the matrix. The 8279 debounces the keys before they are entered into the FIFO. When a key is found, the 8279 checks it two scans later. If it is still depressed, then it will be entered into the FIFO. The scan cycle is programmable. Intel recommends 5.1 ms giving a 10.2 ms debounce time. The CPU will be interrupted whenever the FIFO is not empty if the INT line is used.

There are also two submodes. In 2-key rollover, only a single key can be pressed for an entry to be made into the FIFO. If a key is entered and a second key is pressed while the first key is still depressed, the second key will not be entered until the first is released. In N-key rollover, any number of keys can be pressed and all will be entered into the FIFO in the order they are discovered by the 8279 scanner.

The choice between the three modes is fairly straightforward. The Strobed Input mode can be used with a keyboard where the outputs are already encoded, such as

an ASCII keyboard. For most other applications the Scanned Keyboard mode would be the proper choice since it does all the work of debouncing and encoding by row and column. The Sensor Matrix mode does not debounce or encode. Its advantage is that it can determine the duration of a sensor closure and when a key is released. In the Scanned Keyboard mode, an entry is made only for a closure. There is no way to tell if the key is immediately released or held down for some time.

THE 8279 DISPLAY RAM

The Display RAM is 16x8 bits. In the decoded mode, the first four bytes are used. In the encoded mode, a CPU command determines whether the 8279 will scan to seven or fifteen using either half or all of the RAM. The display RAM can be read as well as written, giving the convenience of not having to allocate separate memory in which to store the value of displayed variables. The Display RAM is read or written in the same way as reading the FIFO/Sensor RAM. A command of Read Display RAM or Write Display RAM is issued. The command contains the position where reading or writing will occur. Auto-incrementing can be used with both reading or writing. In addition to a 16x8 organization, the Display RAM can be organized as dual 16x4 memory. One is the A and uses A0-A3 and the other is B and uses B0-B3 outputs. Either half can be masked from writing so that one 4-bit nibble can be changed without disturbing the other nibble. Figure 9 shows how to use this feature with a BCD to 7-segment decoder. This actually doubles the number of digits possible allowing as many as 32 digits on one 8279.

There are two methods available under program control for entry of data into the Display RAM. Under left entry, the RAM positions are numbered from 0 to 15 and correspond to display positions 0 to 15. Auto-incrementing causes sequential positions of the RAM to be filled. Right entry is the method used in calculators of entering from the right and shifting the previously entered digits left to make room. If the Display RAM is used in left entry, it need not be a single number but can be broken into several groups of digits. For example, digits 0-3 could be the address display and 4-5 the data contents. The right mode must be a single display, otherwise, digits would move out of one group and into the next higher group, whenever an entry is made into the Display RAM.

One note of caution in laying out an LED display to be multiplexed by the 8279 — or by any technique for that matter — attention must be paid to the absolute ratings of both the *average* current per segment and the *peak* current per segment. For example, an NSN3881 has a peak current/segment rating of 75 mA and an average current/segment of 20 mA. If sixteen multiplexed display positions are used, to get maximum brightness at 20 mA average would require 20x16 = 320 mA of peak current during the 1/16 of the time when the digit is being turned on, exceeding the peak current. Allowing the current to be at the peak value would give an average of 75/16 = 4.7 mA and a very low intensity display. If the display can be broken so that it can be multiplexed eight digits wide, then the average current would be 75/8 = 9.4 mA providing a much more acceptable intensity. The display could take advantage of the dual feature of Display RAM and be split into 8x4 parts. This would give up to sixteen characters and still limit the multiplexing to eight positions.

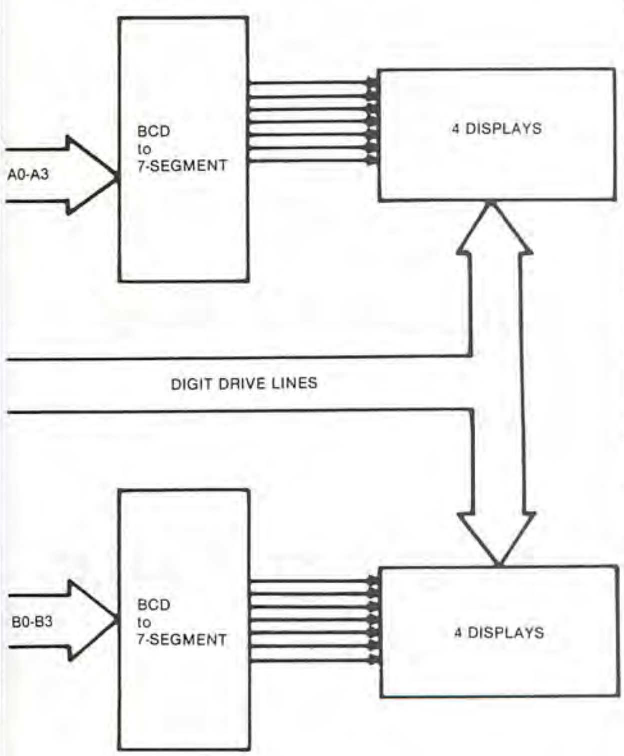


Figure 9. Using display RAM as dual 4-bit displays.

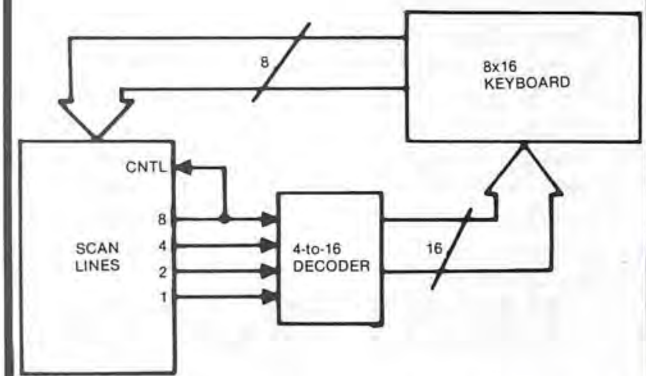


Figure 10. Expanding to an 8x16 keyboard.

CLARIFICATION OF 8279 SPECIFICATIONS

Several items in the Intel specs are incomplete or confusing. Here is my interpretation of what was meant.

Intel claims that the 8279 in the Scanned Keyboard mode can handle an 8x8x4 keyboard encoded and 4x8x4 decoded. The final 4 refers to the fact that there can be four conditions under which a key is entered into the FIFO. They are unshifted, shifted, control, and control/shift. Remember that the SHIFT and CONTROL keys are not part of the key matrix but have separate input lines whose value is put into the FIFO with the key position.

The specs claim that the 64-key matrix is expandable to 128. This can be done by using an 8x16 matrix and feeding the most significant bit of the scan lines into either SHIFT or CNTL/STB (Figure 10). The 8279 is pro-

KEYBOARD INTERFACE	DISPLAY INTERFACE
R0-R7 — Return inputs	A0-A3 — Display (A) outputs
CNTL — Control input	B0-B3 — Display (B) outputs
SHIFT — Shift input	$\overline{BD}$ — Blank display output
S0-S3 — Scan outputs	S0-S3 — Scan outputs

Table 2. 8279 Interface Signals.

grammed for Scanned Keyboard. Note that debouncing still functions properly because the debounce logic ignores one scan (eight counts) and then looks again to see if the key is still pushed. By thinking of the keyboard as two 8x8 sections, a key push in section 1 will not be debounced using the corresponding key in section 2. Each key is only looked at every other scan because the 8279 believes it is scanning an 8x8 matrix.

The output pin labeled $\overline{BD}$ is used to blank the display. The reason for this is not clear in the specs. It turns out that during multiplexing, the 8279 takes a relatively long time to switch the output display drivers (up to 100 μs). The $\overline{BD}$ output changes state while the output lines are in transition. If it is not used to blank the display, ghost images will appear on the LED segments that get turned on during the digit switching interval.

Finally, the specs neglect to explain how the A and B display output are taken from the data bus lines. The correct order is:

D7	D6	D5	D4	D3	D2	D1	D0
A3	A2	A1	A0	B3	B2	B1	B0

KEYBOARD/DISPLAY APPLICATIONS

Interfacing a keyboard and a display to a microprocessor is clearly superior to terminals in many instances. A microprocessor monitor front panel with a button labeled INSERT is simple to use. On the other hand, when faced with a new keyboard monitor, is the terminal command to put something into memory as I for insert or A for alter or C for change. Even a BASIC keyboard would not be so far fetched with a pushbutton for each statement type (e.g., LET, IF, THEN, GOTO, GOSUB, FOR, REM, PRINT, NEXT, etc.). That should speed up your BASIC program development and reduce key errors. (Oh, how about an ERASE LINE Key while you're at it instead of Control/Shift something). The BASIC keypad could augment a regular terminal, provided you have access to the BASIC source. Could this be extended to assembler with a key for each mnemonic?

A game aficionado can design a general purpose keyboard display to be used with many different games. It should have easily changeable labels for keys and displays. Think of a Star Trek Consol with all the functions on keys (start warp engines, fire photon torpedoes, etc.) and critical parameters flashed on the displays (ships clock for stardate, energy levels, etc.). Certainly many CRT's would benefit from a cursor control keypad, a numeric keypad or an editing keypad. In addition, the Editor program could use a supplementary keypad.

For music generation, wouldn't an 8279 be able to handle your keyboard requirements? A keyboard/display should just do the trick for that stand-alone controller you were thinking about for your home heating/cooling systems.

CONCLUSIONS

I have attempted to show that interfacing keyboards and displays to a microprocessor was never very difficult and is even easier now that a keyboard/display peripheral exists. I have also tried to illustrate with my list of possible applications that in many areas there are cheaper and better ways to do things than using a CRT or a terminal. I think the word better should be emphasized because the keyboard/display in many cases is better suited and more convenient than anything else, no matter how expensive.

Finally, I hope I have convinced you that custom-designed keyboard/displays are as valid and desirable in projects where only a single unit will be put together as in mass-produced electronic products.

REFERENCE

Intel Corporation, *MCS 85 User's Manual,* Preliminary, Section 5, pp. 5-13, 5-23, March 1977.

AUTHOR'S NOTE

Larry Grim has been working with computers of various types for more than 10 years, first at graduate school in electrical engineering and then for the last eight years working for E.I. DuPont in Deepwater, New Jersey. He is also a Ph.D. candidate in computer science at the Moore School of Electrical Engineering at the University of Pennsylvania. Grim has worked on many different machines and has written programs in FORTRAN, APL, COBOL, PL/I, BASIC, and many assembler languages. In 1977 he was primarily involved in microprocessor projects at DuPont.

"I wish I could be more helpful, Sir, but I see no computer industry standards coming up."

VideoBrain
The Consumer Computer

By Mike Peak

INTRODUCTION

Consumer electronics is one of the oldest industries known as "electronics." Consumer electronics dates as far back as the crystal receiver, and is ever advancing with the state of the art.

During the past three years, a new and exciting generation of consumer-oriented devices has been made available to the American public. These devices have ranged from video games to highly sophisticated video cassette systems. Now as we enter 1978, another new and innovative approach is being taken by the electronics industry.

The industry is now viewing the microprocessor, not only for games, but for viable home computing systems. One company in particular, VideoBrain of Sunnyvale, California, has recently developed one of the more exciting entrants to the new field of consumer computers. This new device is named the VideoBrain.

The VideoBrain is the first true turnkey computer system built with the general consumer in mind. Built around the F8 microprocessor the VideoBrain does not require any computer expertise on the part of the user.

THE CONSUMERS' INTERESTS

According to Dr. David Chung, one of the developers of the system, the VideoBrain was designed to fit five areas of consumer interest.

The first and most obvious is entertainment. The VideoBrain is indeed entertaining. Several off-the-shelf game programs are available with more on the way. The second area is education and self-improvement. With the VideoBrain the user can practice typing or learn music. The third area is personal finance and data storage with hardcopy printouts. The VideoBrain can be used to balance the checkbook or handle the yearly Christmas card list. Fourth, the system is a communication device when used with an audio coupler. Using the VideoBrain as a communications device, the consumer can have the convenience of quering large data bases, or depositing information into them, for use by others. Finally, the area of home control has been addressed. The system, along with the appropriate application package, can be used to handle the daily mundane tasks of controlling the household environment.

Both developers, Dr. Yu and Dr. Chung, feel that the personal computer, up until now, has been in the same class as calculators, chips, and watches. The product is developed and marketed, in large volumes, without sufficient thought going into what the consumer really wants. Therefore, to add viability to the VideoBrain, the broad spectrum of the marketplace has been addressed. Consequently the VideoBrain system provides something for everyone in the family, not just the inhouse computer enthusiast.

BASIC STRUCTURE

The F8 microprocessor, used in the VideoBrain, was originally designed for use as a controller, and consequently does not use the typical hobbyist bus structure. However, the bus structure used by the VideoBrain is designed to allow for the addition of a network of MPUs by simply clipping them on. As Dr. Chung describes it, "The machine uses both a tightly coupled and loosely coupled method of bus architecture. This method of bus design gives the system a great deal of flexibility, when connecting it to other intelligent devices."

. . .it would take 100 4K RAMs, and possibly two MPUs to service the RAMs, to perform the same function.

The system incorporates a new innovative chip, used to handle the screen housekeeping. Therefore, approximately 100,000 points and 16 bright colors can be defined and displayed. To relate this to conventional systems, it would take 100 4K RAMs, and possibly two MPUs to service the RAMs, to perform the same function.

The keyboard used with the system is somewhat unconventional. However, it was designed only after careful consideration of what would be workable for the consumer, and also be cost effective. The keyboard controls all the computer functions including changing the color displays.

The machine in its basic configuration comes with 1k of RAM, which allows for five lines of 16 characters of text. Also included is 4k of ROM for handling the internal working of the machine. The system is expandable, not by adding to the bus, but through the ROM cassettes.

APPLICATION DEVELOPMENT

The area of applications, for the VideoBrain, are by far the most exciting thing about the machine. The applications are designed in the same manner as they would be for a large system. This requires application planning, that is, the need and method are previously defined. Then the entire program development is handled by a programming team utilizing the best of all possible programming techniques. The result is highly reliable and efficient software to handle the needs of the consumer.

The programs are run in the machine by popping a ROM cassette into the built-in reader. Therefore, instant communication exists between the computer and the user.

SUMMARY

The VideoBrain comes with a spectrum of cassette programs from games to personal finance. The plan for future application development is three programs a month selling in the 20 to 50 dollar range, depending upon the complexity of the program.

The VideoBrain people have addressed the problem of add ons and are making available so-called expandables. These expandables will give the user the ability to interface with teletypes, printers, audio cassettes, and sensing devices.

The VideoBrain is designed to be the very first computer system to take full advantage of current technological know-how, and remove the mystery surrounding computers. The VideoBrain provides for man machine interface by utilizing familiar objects found in the home, the color TV and telephone. The basic application design provides something for everyone with the greatest emphasis placed on the homemaker.

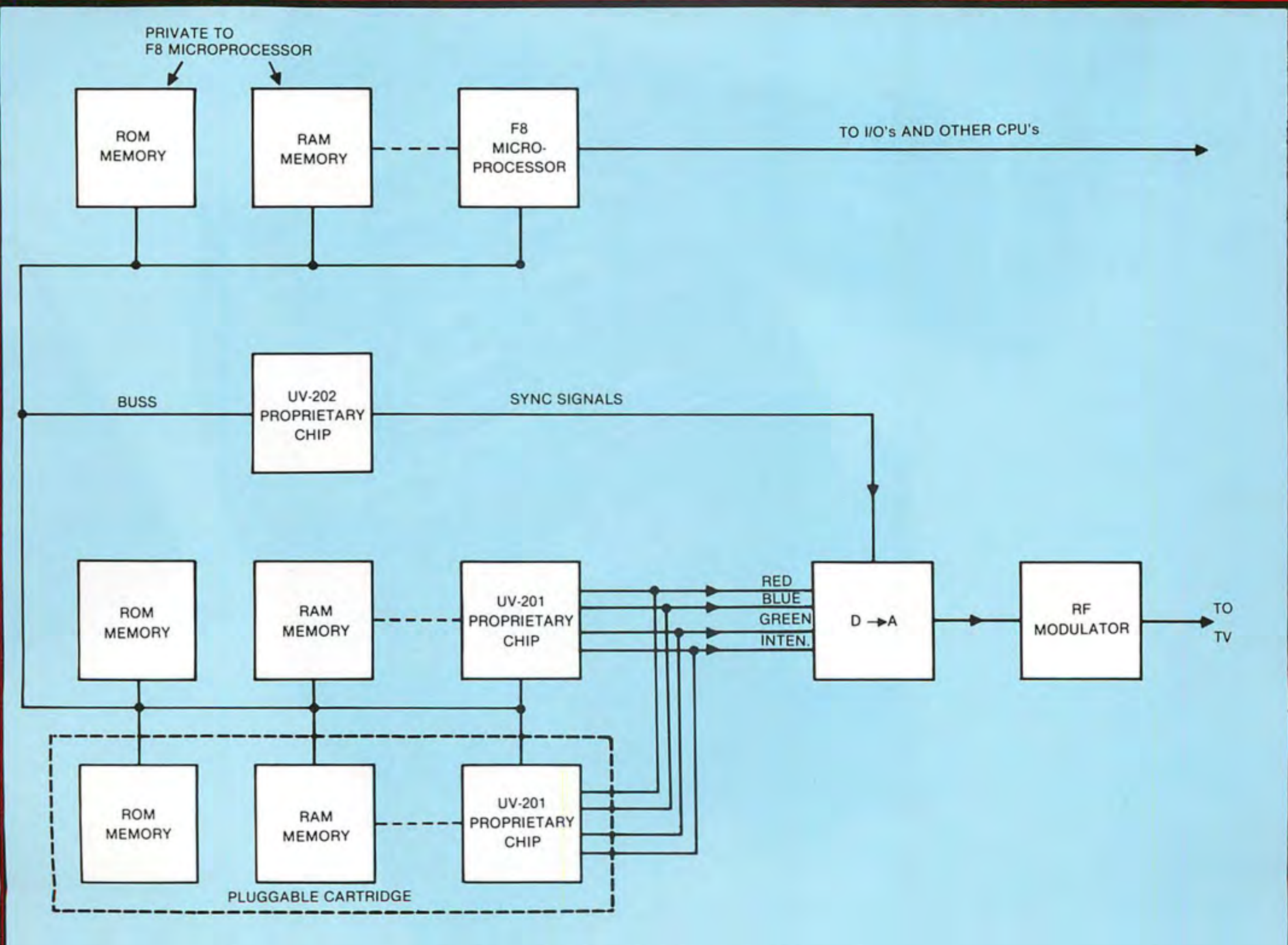


Figure 1. Block Diagram of VideoBrain.

THE VISTA $50 FLOPPY DISCOUNT

We know that one of the biggest problems in personal computing is that you're buying with your own personal dollars.

That's precisely why you're going to like doing business with us.

We're Vista Computer Company, the personal computer systems brainchild of the business computer systems people at Randal Data Systems.

And our V80 Floppy Disk System is a perfect example of how we're prepared to help you get the most out of your personal computing dollars.

$649 buys you the whole kit and kaboodle

The $649 you spend on a Vista V80 Floppy Disk System ($749 assembled) gets you everything you need:

An 80K byte minifloppy drive (assembled and tested) that can be powered directly by your 8080 or Z-80 computer. (Case and power supply optional.)

An I/O cable and a single card, S100 bus-compatible controller kit that handles up to four drives and includes a PROM for bootstrap loading (additional drives just $399).

VOS, the most advanced microcomputer disk operating system available, and our BASIC-E compiler, designed to work with VOS, all on a single diskette. Software functions include instantaneous program loading, named dynamic files, program editing, assembling, debugging, batch processing, and file copying on back-up diskettes.

All backed by the Vista 90-day warranty, membership in VUE (Vista Users' Exchange), and Dataforce, our associated service company with 115 locations throughout the country.

Test drive the V80 at your local computer store

Drop by your nearest computer store and run the V80 through its paces. Once you find out what it can do for you, you'll see that our combination of high performance and low price is hard to beat and easy to take.

We love to take orders

If you'd like us to ship you a Vista V80 Floppy Disk System, they're available now. Just send us a check or money order for the amount of purchase, or your BankAmericard/ VISA or Master Charge account number with expiration date and authorized signature. California residents add 6% sales tax. Uncertified checks require six weeks processing.

To place your order, or to obtain further information, call or write today.

Vista Computer Company, 2807 Oregon Court, Torrance, CA 90503. (213) 320-3880.

Vista

We never forget it's your pocket.

Vectored from Page 65

BUILDING YOUR OWN SIMULATOR

If you are beginning to think that you cannot live without a spaceship simulator in your basement, we will give you some convincing reasons. There is so much to learn about space flight physics, and the mathematics of reentry that killing Klingons is a waste of time by comparison. For example, how many of you realize that when in orbit, firing your engines to go forward will make you go upward as well?

Teachers can use the simulator as a teaching aid for math, physics, and programming. The Walletsize can effectively supplement textbook learning in any of these fields.

The simulator you build need not be this complex. On the other hand, maybe you can expand it into a three seater with holographic displays. Feel free to adapt any of our ideas into your own configuration. We have sets of the woodworking blueprints available. If you want to correspond with us about the ship, our address is: 2005 Naudain Street, Philadelphia, Pennsylvania 19146.

One final word of warning — this project will be enormously time consuming. So unless your family can exist without you for long periods of time, you'd better teach them all to help.

DESIGN CRITERIA

The physical design criteria for the ship required that it had to be small enough to fit easily into our workshop, yet large enough to comfortably accommodate at least one person. The materials had to be inexpensive and readily available, and the entire unit had to be portable.

The final size of all the component parts was determined through careful use of the Dreyfus tables, available at architect and builder's stores. These tables showed us the average sizes of human beings, the range of motion which could be comfortably expected from arms and legs, and the proper height for the chair (Figure 14). Every part of the simulator was designed with the average-sized human operator in mind. The finished box was later tested with both larger and smaller than average operators, and was found to be more than adequate.

Another factor was that all parts must be able to be produced from standard size lumber, using standard fittings (see Table 3). Since we wanted to construct the ship in our workshop, then move it to the office, and possibly display it elsewhere, it was decided to break the simulator down into sub-units which could then be reassembled at the site. With this in mind, we selected a bolt and tee nut system which would simplify assembly and disassembly.

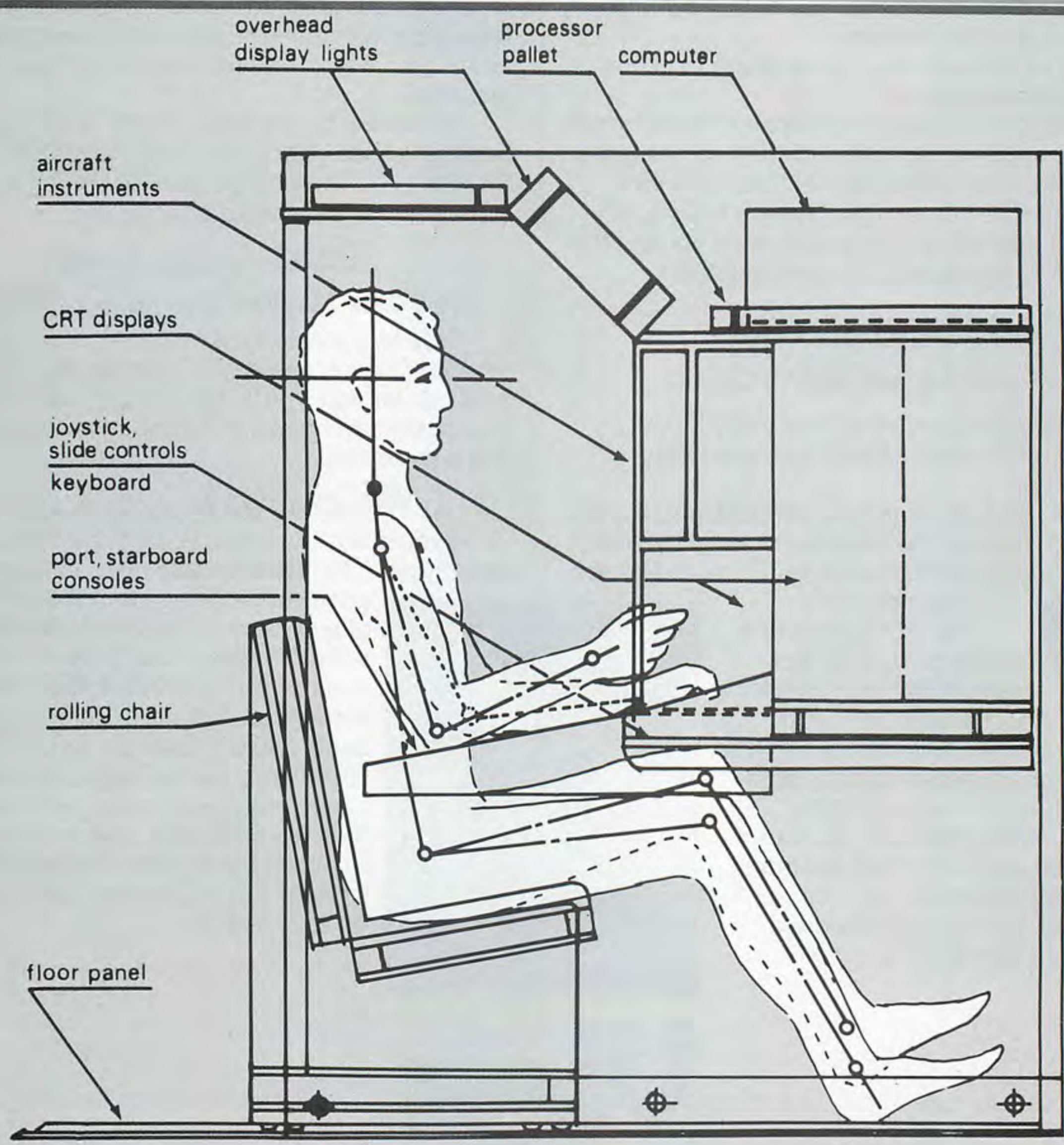


Figure 14.

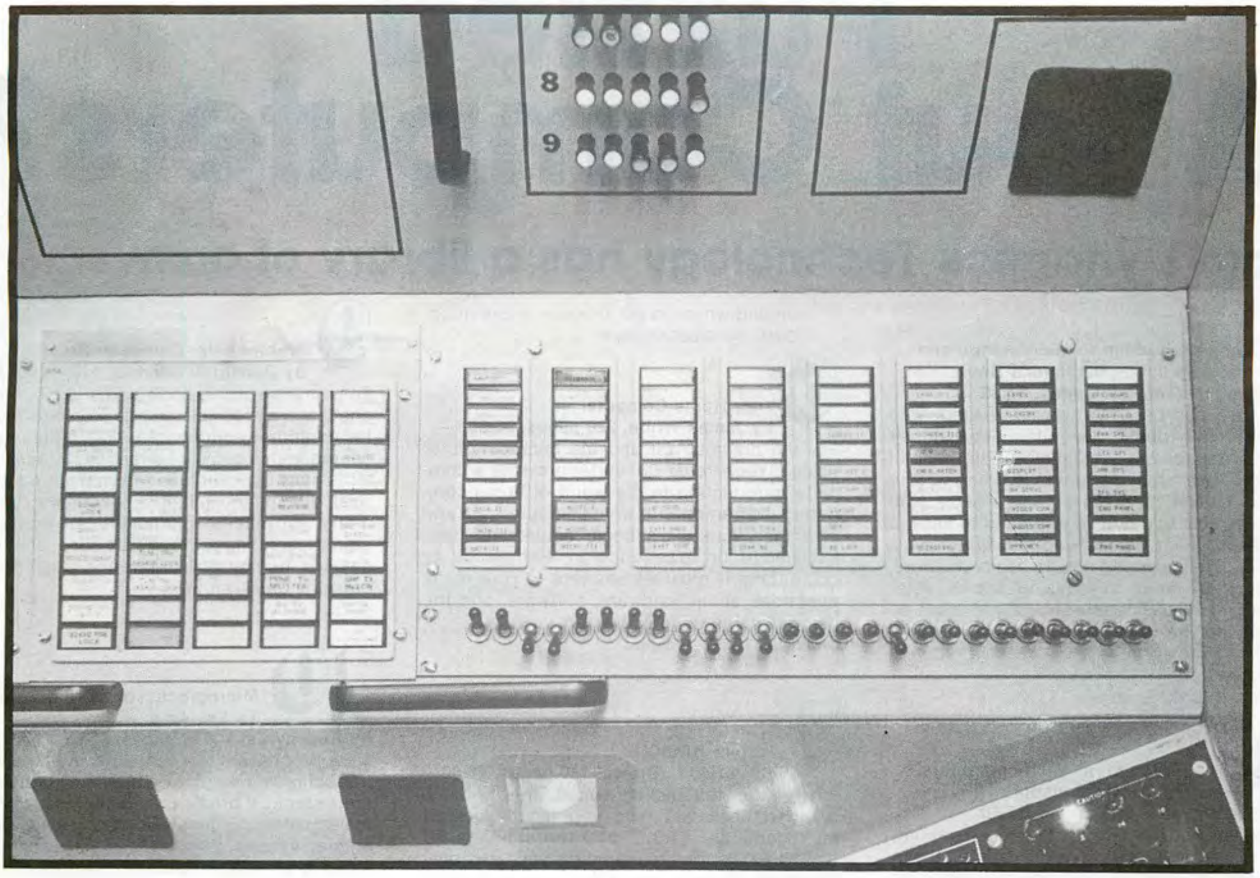


PHOTO 8 Items shown (left to right): wire harnesses and trimmed plastic boxes, aircraft surplus, cardboard and spare parts homebrew display.

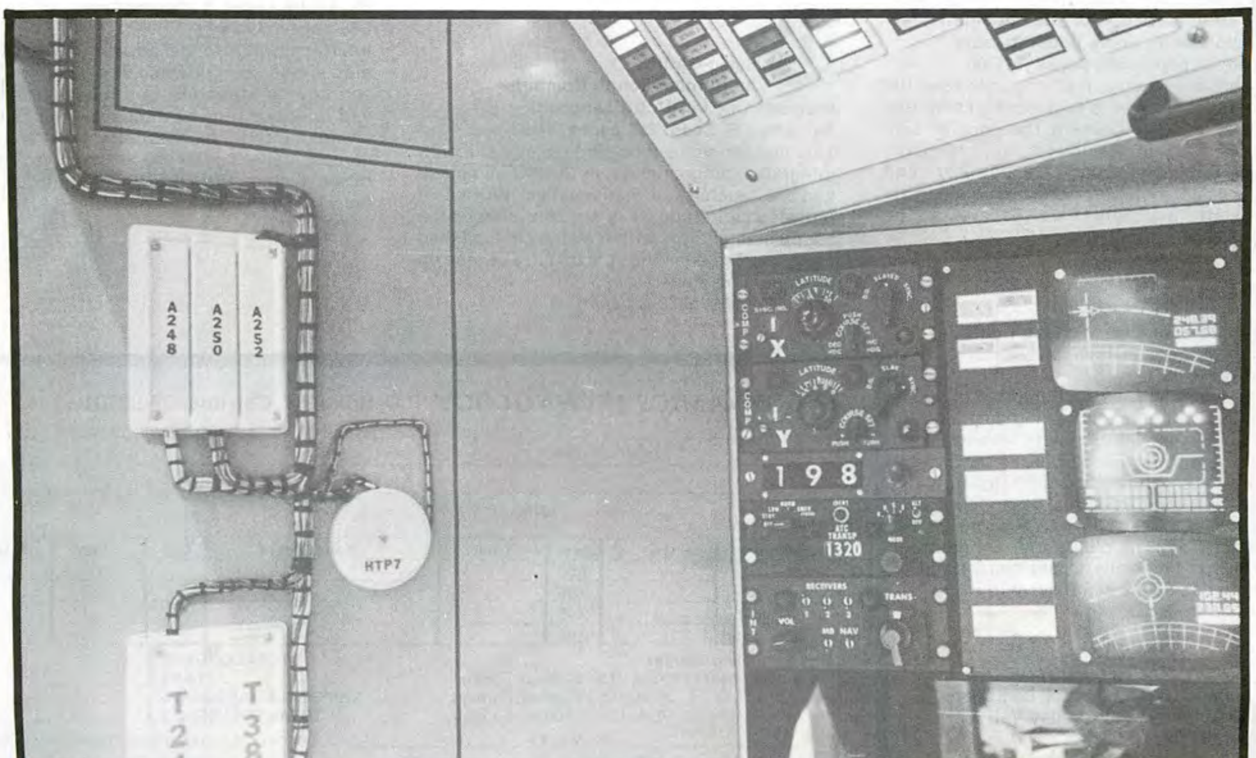


PHOTO 9. Surplus switches and Apollo surplus displays.

Puzzled About Computers?

Data Dynamics Technology has a library of answers . . .

1

An Introduction to Microcomputers: Volume 0, The Beginner's Book
By Adam Osborne. 226 pages, $7.50
This book has been written for readers who know nothing about computers, and it has been written for two audiences: 1) For those of you who have a real interest in learning how to use computers. 2) For everyone else — who must live with computers, and need to know a little about them. For those of you who have a real interest in learning how to use computers, this is the first in a series of books. After reading this book you will be ready to move on to "Volume 1 — Basic Concepts," which gives you the information you will need in order to use computers.
Order No. OSB6001, paper.

2

An Introduction to Microcomputers: Volume 1, Basic Concepts
By Adam Osborne. 287 pages, $7.50
The purpose of this book is to explain not only what microcomputers are, but in addition, why they must be evaluated in a way that differs so markedly from prior computer comparisons. The book does not assume you understand how computers work; therefore, computer concepts are described beginning with first principles.
Order No. OSB2001, paper.

3

An Introduction to Microcomputers: Volume II, Some Real Products
By Adam Osborne. 868 pages, $15.00
As the second volume in a two-volume set, the purpose of this book is to describe some real products which implement the general concepts covered in Volume I. In this book, devices of the 8080A, MC6800, Z80, and MCS6500 microcomputers are described in approximately the detail we believe to be necessary.
Order No. OSB3001, paper.

4

Build Your Own Working Robot
By David L. Heiserman
234 pages, $5.95
Here are complete instructions, plans, schematics, logic circuits, and wiring diagrams for building Buster, the most lovable (and mischievous) mechanical pet in the world! He'll serve you coffee or bring you the morning papers. He'll forage for his own "food" and scream when he can't find it. His "curiosity" will get him into one plight after another, but Buster has the capacity to get himself out of trouble just as easily as he got into it!
Order No. TB841, paper.

5

Getting Involved with Your *Own* Computer: A Guide for Beginners
By Leslie Solomon and Stanley Veit
216 pages, $5.95
This is the first major book by outstanding authorities that assumes only that you have an interest in what makes personal computers work and what can be done with them. It will not make you a computer expert, but it will teach you enough to understand what is going on and where to go for more information.
Order No. RID004-8, paper.

6

Your Home Computer
By James White. 220 pages, $6.00
For the pre-hobbyist and the microcomputer novice, *Your Home Computer* provides a complete introduction to the world of home computing, beginning with what computers are and how they work. This book requires no prior knowledge or experience in electronics or computing. It provides answers to your many questions about hardware, software, and the personal computing scene today.
Order No. DMX 05-1, paper.

7

Instant BASIC
By Jerald R. Brown. 180 pages, $6.00
Written for the inexperienced, this activity-oriented book will help you teach yourself microcomputer BASIC, and the similar DEC BASIC PLUS for programming your personal computer. There's never a dull page and plenty of activities . . . so have fun while you learn! For those of you who already know some BASIC, this workbook can teach you the new microcomputer dialect.
Order No. DMX 04-3, paper.

8

Basic BASIC: An Introduction to Computer Programming in BASIC Language
By James S. Coan. 256 pages, $8.95
This text for high school and college students integrates programming in BASIC language and the teaching of mathematics. Written in precalculus setting, it is suitable either as a supplementary text within already established math courses or as the sole text in a course on programming alone.
Order No. HAY5872-1, paper.

9

Problems for Computer Solution
By Donald D. Spencer. 125 pages, $6.95
The purpose of this book is to bring together in one place a wide selection of problems suitable for computer solution. The 821 problems presented in this book are arranged by topic. Problems have been included from most mathematical disciplines (algebra, geometry, trigonometry, advanced mathematics, probability, statistics, number theory), science, chemistry, physics, business, biology, and game playing.
Order No. CAM019-6, paper.

10

Microprocessors: From Chips To Systems
By Rodnay Zaks. 416 pages, $9.95
Microprocessors is a complete and detailed introduction to microprocessor and microcomputer systems. It presents both the concepts, and the actual techniques and components used to create systems. It introduces the reader to the aspects of system operation, use, and design.
Order No. C201, paper.

11

Microprocessor Interfacing Techniques
By Austin Lesea & Rodnay Zaks.
348 pages, $9.95
Interfacing is no longer an art, but a set of techniques and components. This book will teach you how to interconnect a complete system, and interface it to all the usual peripherals. It covers hardware and software skills and techniques, including the use and design of model buses such as the IEEE 488 or S100.
Order No. C207, paper.

DATA DYNAMICS TECHNOLOGY P.O. Box 1217, Cerritos, CA 90701

Name(Print) ____________________

Address ____________________

City ____________________ State ________ Zip ________

Please send me:

Description	Qty	Price	Total	Book Order #	Qty	Price	Total
Binders		7.50					
Slip Cases		5.95					
Hex Code Cards		2.95					
Octal Code Cards		2.95					

Shipping & Handling Charges
Binders and Slip Cases $1.00 ea. U.S., $1.50 ea. Foreign
Code Cards $ .25 ea. U.S., $ .50 ea. Foreign
Books $ .50 ea. U.S., $1.00 ea. Foreign

TOTAL ORDER $ ________
***TAX $** ________
SHIPPING & HANDLING $ ________
TOTAL ENCLOSED $ ________

☐ Visa # ________ ☐ M/C # ________ ☐ Check or M.O.(U.S. Funds drawn on U.S. bank)

Exp. Date ________ Signature ____________________

*California residents add 6% sales tax. Prices quoted subject to change without notice.
Please allow six weeks for delivery. You may photocopy this page if you wish to keep your INTERFACE AGE intact.
Orders *cannot* be shipped unless accompanied by payment, including shipping & handling and tax where applicable.

DATA DYNAMICS TECHNOLOGY, A Division of INTERFACE AGE Magazine (213) 926-6629

1


2


3

4

5

12

TYCHON's 8080 Octal and Hex Code Cards

The code cards are a sliderule-like aid for programming and debugging 8080 software. Both cards contain all the standard mnemonics and either their corresponding octal or hex codes. The pocket size cards are 6.5 by 3 inches (16 by 8 cm) with color-coded instructions to provide a neat, logical format for quick reference. The back of both cards is printed with an ASCII code chart for all 128 characters plus the 8080's status word and register pair codes.

13

INTERFACE AGE Binders and Slip Cases

Collecting magazines can bring headaches — not to mention dust, ripped pages and misplaced copies. If you use your back issues of INTERFACE AGE as reference material, nothing is more annoying than taking time to find mislaid copies. Data Dynamics Technology is now offering deluxe binders and slip cases which will place each back issue of INTERFACE AGE at your fingertips. Each binder and slip case is constructed of a handsome blue vinyl with INTERFACE AGE stamped in gold foil on the front cover and spine. These rugged binders and slip cases can hold 12 issues each and will protect your back issues of INTERFACE AGE for years.

Instant BASIC
By Jerald R. Brown. 180 pages, $6.00
Order No. DMX04-3, paper.

Basic BASIC: An Introduction to Computer Programming in BASIC Language
By James S. Coan. 256 pages, $8.95
Order No. HAY5872-1, paper.

Beginners Guide to Computer Logic
By Gerald F. Stapleton. 192 pages, $7.95
Order No. TB548, cloth.

The Systems Analyst: How to Design Computer-Based Systems
By Jerry T. Atwood. 225 pages, $10.95
Order No. HAY5102-6, cloth.

Accent on BASIC
By Donald D. Spencer. 104 pages, $5.95
Order No. CAM003-X, paper.

Microprocessor Programming for Computer Hobbyists
By Neil Graham. 382 pages, $8.95
Order No. TB952, paper.

Simplified Computer Programming — The Easy RPG Way
By Kelton Carson. 240 pages, $5.95
Order No. TB676, paper.

Problem Solving with FORTRAN
By Donald D. Spencer. 320 pages, $12.40
Order No. PH0094-3, paper.

Scientific and Engineering Problem-Solving with the Computer.
By William Ralph Bennett, Jr. 457 pages, $22.30
Order No. PH5807-2, cloth.

Techniques of Program Structure and Design
By Edward Yourdon. 364 pages, $21.10
Order No. PH1702-X, cloth.

BOOKS ONLY
For Bulk Store Quantities contact:
MicroMedia Marketing, (213) 681-0685

Prices quoted subject to change without notice.

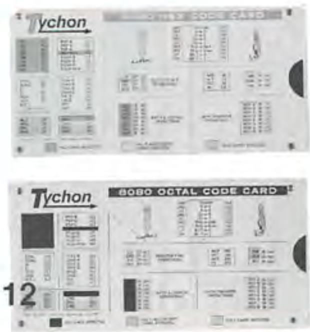
6

7

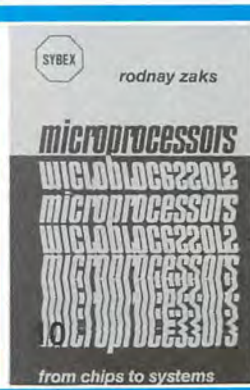
8

9

13



12



10

11

CIRCLE INQUIRY NO. 13

PHOTO 10. A pile of spaceship.

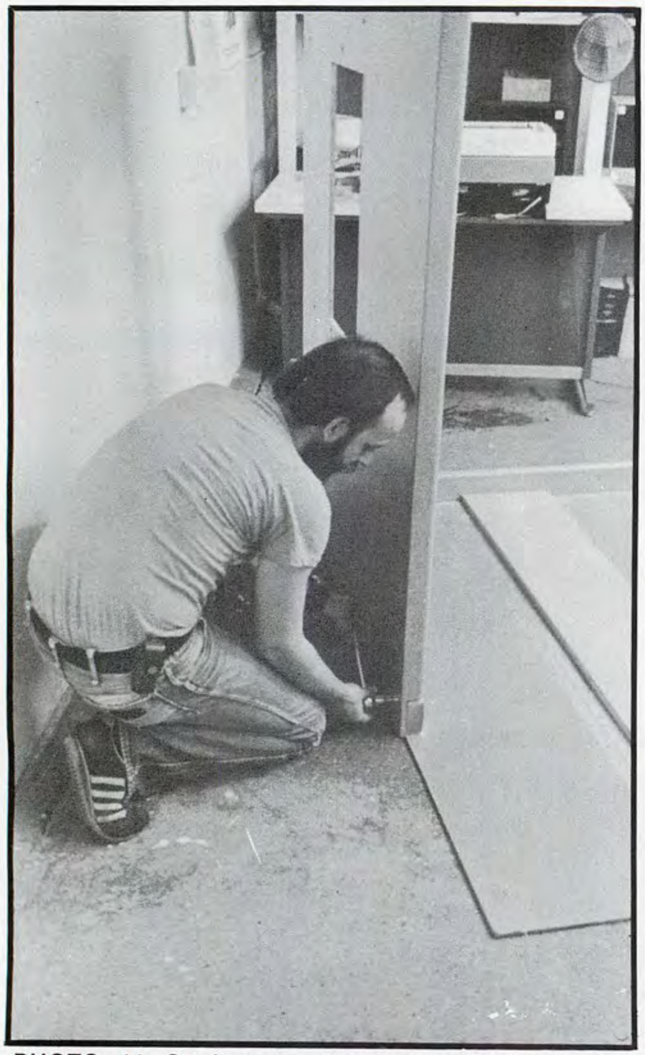


PHOTO 11. Starboard cabin wall bolted loosely in place.

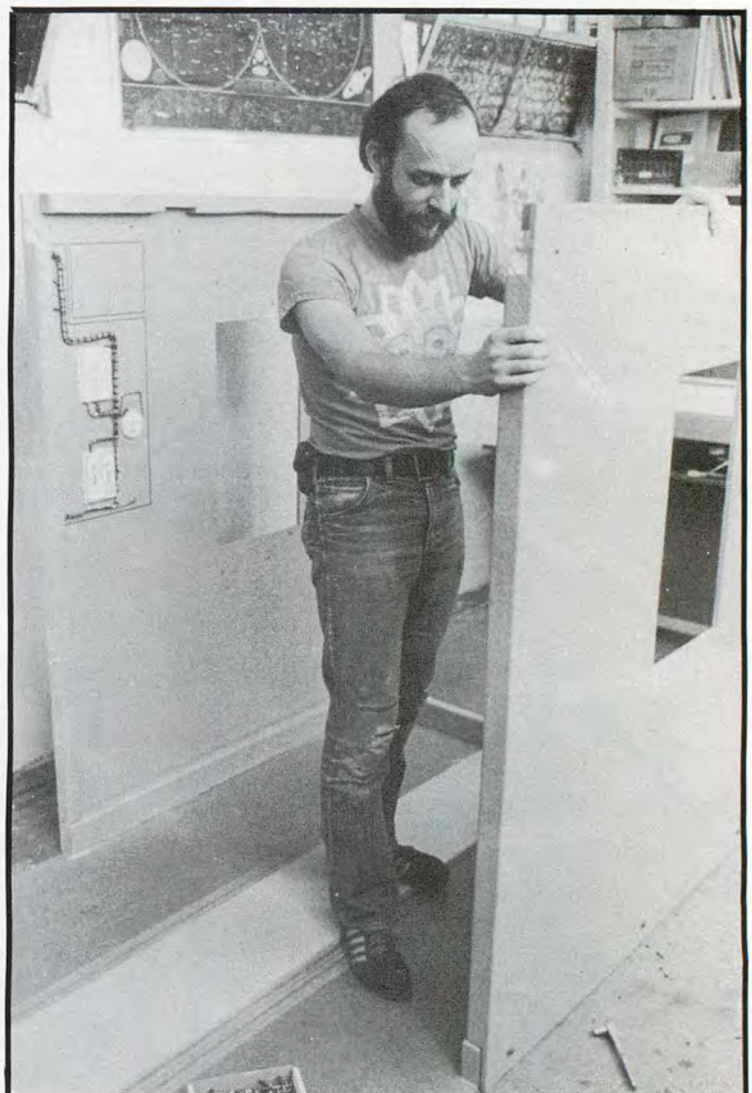


PHOTO 12. Port cabin wall positioned.

A third factor, which affected the size and design of the ship, was the type of surplus display equipment which we found to be available. Consequently, we designed the overhead display panel to accommodate standard 19" rack-mount equipment. The small instrument boxes were designed to hold surplus aircraft instruments and small monitors. Other panels were custom made to hold the smaller displays, such as the momentary switches on the port console, the joystick and slide controls, and the overhead circuit breaker panels. Wood, metal and cardboard were used for these panels. the panels were painted, or covered with adhesive backed vinyl to simulate a painted metal finish.

A complete set of working plans were drawn to scale. The plans were turned over to our "shop steward," who then collaborated with the woodworking shop. The five blank panels and 6 plain 2x4s were then turned into a stack of finished pieces. All pieces were sanded, and all holes or cracks filled in with spackle.

The pieces were then given two coats of battleship

PHOTO 13. Instrument pallet.

PHOTO 14. Overhead display frame being passed through the opening in the cabin wall.

PHOTO 15. Bolts loosely screwed down as the overhead display frame is held in position.

gray polyurethane paint. This type of paint was chosen because it holds up well under hard use, and tended to make the plywood look more like metal. The paint is somewhat expensive, but is well worth it. Most pieces were painted on all sides to eliminate the possibility of warping.

The sub-units, such as the overhead lighting displays and the instrument boxes were then permanently assembled. Lighting displays were bolted into place and electrical fixtures were installed. Speakers and overhead panel lights were plugged in, and some detailing was added with narrow black drafting tape. On the side panels, which looked conspicuously blank, we attached small white plastic boxes trimmed with colored stick on numbers; wire harnesses were glued between these for a "NASA" effect (Photo 8). The small side boxes were fitted up with various aircraft instruments, and colored film display panels, and then fitted into place.

DETAILS OF THE DISPLAY

The overhead telltale light panels were surplus from the Apollo space program; they contain a total of 186 lights. At present a limited number are tied into the processor, with the remainder being worked from switches at the right side of the panel.

The port instrument box contains several pieces of surplus aircraft instrumentation, as well as three handmade displays. The starboard box contains aircraft instrumentation for the inner workings of one of these displays. This includes a destruct switch, and a mine dispenser. A small surplus monitor is housed in the base of each box and is covered with smoked plexiglas. A 25" (diagonal) high-resolution monitor, also bought at a surplus house, serves as the main display.

The port console contains three slide potentiometers used to control the ship's ascent and descent, and an array of lit momentary switches backed by color chips. These controls are set into a handmade panel enriched with transfer type labels. The starboard console holds the joystick, which makes three dimensional flight possible, plus another switch panel. The technical details of these controls are fully explained in part two of this article. Both consoles are fitted out with padded armrests, as well as black metal hand grips.

PHOTO 16. Cabin top panel is attached in the same manner.

STOP!

Before you write that letter to the editor, renew your subscription, or send in a new one; or forward any inquiries or correspondence to us, we would like you to know we have moved.

Our new address is:

INTERFACE AGE Magazine
16704 Marquardt Ave.
Cerritos, CA 90701
(213) 926-9544

Two 7½ watt bulbs were mounted into the ceiling, to aid visibility, and six 2½" speakers were mounted inside the ceiling and consoles. These speakers are coupled to the analog devices and to the processor, to give audible clues to the handling of the ship.

The rolling chair was made by covering polyfoam padding with canvas and fitting it to the wooden chair pieces. The chair was then given a set of refrigerator dolly wheels, which enabled it to roll along a track on the floor. A small metal bracket at the extreme back end of the floor panel prevented the chair from rolling off.

ASSEMBLY PROCEDURE

Photo 10 shows the spaceship in a pile. Several items shown in the foreground are not seen in the final assembly. These items include: The two small monitors and keyboard which form the operator's console on the processor pallet, and the cassette machine with 3 tapes containing programs required to make the Walletsize functional. Photos 11 and 12 show the cabin walls in place and bolted loosely, allowing play for the insertion of other parts. The instrument pallet is brought in sideways, and then bolted loosely in place beneath the openings in the cabin walls. The construction details of the instrument pallet (Photo 13) is the key to the unit's strength.

The processor pallet is slid inside the walls through the space at the top forward corner of the cabin, and then dropped in place over the side wall openings. The port and starboard consoles are held loosely in place by three bolts from the outside and two on the other side of the console, through the pallet under the keyboard. The keyboard is laid in place and the bolts underneath it are tightened, pulling the port and starboard consoles in and holding the keyboard in place. Two bolts are inserted through the instrument pallet into the back of each con-

sole and tightened down. These bolts are hidden by the instrument panels.

The overhead display frame (Photos 14, 15) is passed through the openings in the cabin's walls and held in place while bolts are put into place from either side. The cabin top panel (Photo 16) is slid into place and held likewise. The port and starboard instruments (Photos 17, 18) are wedged into the openings in the walls and two bolts are finger tightened at the top and bottom of each box.

The monitor is then put into place (Photo 19) and snugged into position between the two instrument panels. With most of the pieces now in place, the sides and the floor are held square and the bolts going through cabin walls are tightened down. The bolts going through the top and bottom of the instrument cases are then tightened (Photo 20).

The modesty panel (Photo 21) is put into place and the cover (Photo 22) fastened to its hinge. The processor (Photo 23) is put into place on top of the processor pallet, and wiring begins (Photo 24). As the simulator is being wired, the chair is assembled and rolled onto its tracks (Photo 25). The result is the completed SFS Walletsize Space Simulator.

PHOTO 17. Starboard instruments slid through opening in cabin wall.

CALL FOR ARTICLES

Articles authored by individuals during leisure time are remunerated at a rate from $15.00 to $50.00 per published page and articles describing company projects carry author and company byline, but no honorarium is offered. Articles accepted will be acknowledged with a binder check within 30 days of receipt.

Manuscripts should be double-spaced, typewritten pages, one inch margins, and not less than 3½ pages in length (one published page). Pages should be numbered to insure correct text. Photographs should be numbered and labeled on the backside with a description. Tables, listings, etc., shall be on separate sheets. Photos should be taken with uniform lighting and background, in the form of glossy black and white prints. Computer listings shall be printed using a new ribbon to assure darkest print copy. Authors shall supply a statement of their background, expertise and level of accomplishment.

The publisher assumes no responsibility for artwork, photos, models, or manuscripts. Manuscripts are not acknowledged or returned unless accompanied by an addressed, stamped, return envelope.

For article submittal or further information, contact respective editor, INTERFACE AGE Magazine, 13913 Artesia Boulevard, Cerritos, CA 90701 or call (213) 926-6629.

PHOTO 18. Two top bolts located through processor pallet into port instrument cases.

PHOTO 19. The monitor snugged into position.

PHOTO 20. Bottom bolts of port instruments tightened.

PHOTO 21. Modesty panel in place.

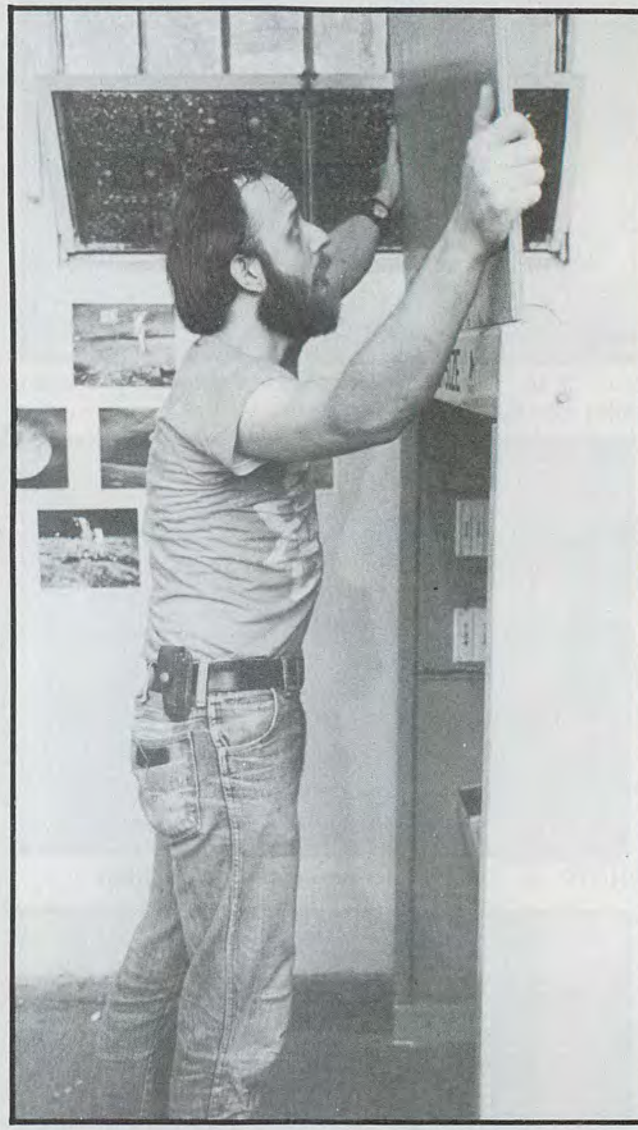


PHOTO 22. Cover fastened to its hinge.

PHOTO 23. Processor positioned on top of processor pallet.

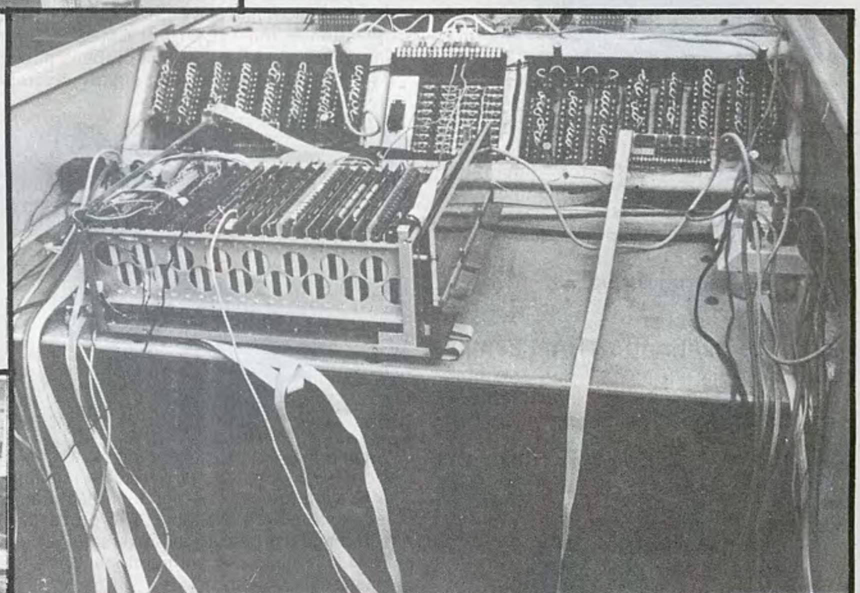


PHOTO 24. Wiring!

PHOTO 25. Chair slid onto track following assembly.

NEW PRODUCTS

Electrically Conductive Bags for Electronics Packaging

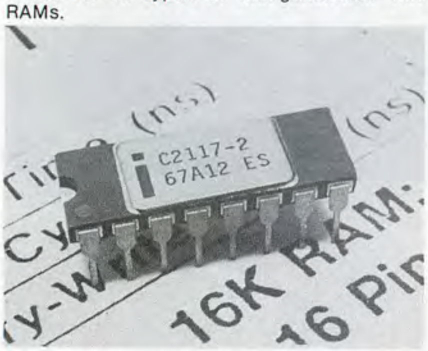
Charleswater CP302 Statfree Conductive Bags are electrically conductive durable non-woven nylon bags that protect CMOS, MOS, printed circuit boards and other delicate electronic parts from static electricity. CP302 bags are reusable. They are 5 mils thick and surface resistivity is less than 30 x 10^3 ohms/sq.

Charleswater CP302 Statfree Conductive Bags are priced from $21.50 per 100 bags. Quantity discounts and a complete price schedule are available. Custom imprinting costs $2.00 per 100 bags, 1000 minimum. For more information contact Charleswater Products, Inc., 3 Walnut Park, Wellesley, MA 02181, (617) 237-5942.

CIRCLE INQUIRY NO. 112

Second Generation 16K RAM Family

Designated the Intel® 2117, this new high performance family provides direct replacements for all types of first-generation 16K RAMs.



The 2117 normally operates as an unlatched-output RAM, with conventional multiplexed address inputs, RAS and CAS. Strobe timings are not critical, allowing high performance to be maintained at the system level.

Prices in 100-up quantities are: 2117-2 (150 ns maximum access), $150.00; 2117-3 (200 ns maximum access), $41.00; and 2117-4 (250 ns maximum access), $39.00. All selections in the family are currently supplied in ceramic packages. For more information contact Intel Corporation, 3065 Bowers Ave., Santa Clara, CA 95051, (408) 249-8027.

CIRCLE INQUIRY NO. 113

Artec Offers Free Catalog

A six-page, two-color brochure describing a broad line of microcomputer boards, printed circuit boards and off-the-shelf designer breadboards is available from Artec Electronics, Inc.

The Artec catalog, which also contains a section on the firm's technical capabilities, highlights such products as the 32K byte static memory board (32K-100) designed for both microcomputer hobbyists and small business applications.

Also featured are an S-100 compatible general-purpose board (GP-100), a wire-wrap breadboard (WW-100), and specifications and pricing on a complete line of general-purpose breadboards.

The capabilities section shows with photographs the precision engineering steps taken to manufacture each Artec board.

For a free copy of the brochure, write or call Artec, 605 Old Country Rd., San Carlos, CA 94070, (415) 592-2740.

CIRCLE INQUIRY NO. 131

LP-2 Logic Probe

The new LP-2 low cost logic probe from Continental Specialties Corp., offers performance and features rivaling those of units many times its low $24.95 suggested price.

Separate dual threshold window comparators drive the HIGH and LOW LEDs. A switch selects operation in the TTL/DTL or CMOS/HTL families. Since the LP-2 low cost probe is powered from the board under test, supply discrepancies just can't creep in.

The LP-2 power connector is a standard phono jack. A mating power cable is supplied with alligator clips; other manufacturer and user options are available. The probe is protected against power lead reversal. Probe tips are also replaceable, with optional tips and accessories available.

For more information contact Continental Specialties Corp., 44 Kendall St., New Haven, CT 06509; (203) 624-3103.

CIRCLE INQUIRY NO. 127

Protect Your SOL Computer

An attractive and durable blue denim dust cover especially tailored for the Processor Technology Corporation SOL Terminal Computer is being offered by Curtis Electro Devices, Incorporated.

Designed to protect the unit from dust and little fingers, the cover comes in two standard versions. The first is embroidered with the legend "SOL COMPUTER," the second is lettered "MY COMPUTER." Custom legends are also available for a $5.00 additional cost. The standard cover is priced at $14.95 postpaid from Curtis Electro Devices, Inc., Box 4090, Mountain View, CA 94040.

CIRCLE INQUIRY NO. 116

Mechanical Differentials Offer Low Breakaway Torque

Harowe differentials are relatively simple devices that can perform important functions, especially in computing instruments by adding or subtracting two variables.

Six basic sizes are available with swing diameters from .562 to 1.312. Units feature stainless steel construction throughout including bevel gears; ball bearings are ABEC5; backlash as low as 15 minutes; speeds up to 1,000rpm; breakaway torque as low as 0.1 in.-oz., and operating loads up to 50 in.oz.

End gears are supplied to customer requirements. For more information contact Harowe, Planet Instrument Div., P.O. Box 547, West Chester, PA 19380. (215) 692-2700.

CIRCLE INQUIRY NO. 123

5¼" Floppy System Provides Million Byte Storage

The first low-cost 5¼" floppy disc system offers more than one million bytes of storage and is built to exceed the reliability standards previously associated with larger conventional 8-inch floppy drives.

Designated the MetaFloppy™ 1054 Mod II, the system comprises four drives in a dual/dual configuration, a controller, power supply, chassis, enclosure, all cabling, and a new BASIC software package.

The 1054 will plug into any 8080-based or Z-80-based microcomputer using an S-100 bus, including Altair 8080, COMPAL-80, IMSAI and similar systems. Each 1054 can store a total of 1,260,000 8-bit bytes.

Single-unit price is $3,220. Current delivery on new orders is 45 days ARO. For more information contact Micropolis Corp., 7959 Deering Ave., Canoga Park, CA 91304, (213) 703-1121.

CIRCLE INQUIRY NO. 140

Diskette-Based Intelligent Mass Storage Systems

A new series of IBM compatible Mass Storage Systems for minicomputer applications has been introduced by PerSci, Inc.

The new PerSci systems, with data capacities to 1 Mbyte formatted, incorporate PerSci's Model 277 Dual Diskette Drives and are available in a variety of configurations: (A) A one or two drive (two or four spindle) system complete with microprocessor based controller, power supply and cabling enclosed in a 19" rack mountable chassis; (B) A one or two drive system with power supply and cabinet but without the controller; and (C) A "slimline" ysstem, measuring only 4½" wide when vertically mounted, which incorporates one dual drive (two spindles) and a power supply in a table top chassis.

For further information contact PerSci, Inc. through B.J. Johnson & Associates, 2503 Eastbluff DR., Suite 203, Newport Beach, CA 92660, (714) 644-6037.

CIRCLE INQUIRY NO. 154

VideoBrain™, the First Entry Level Family Computer

VideoBrain, the first home computer that requires no computer programming skill to implement high level computer tasks.

The latest generation product utilizes a preprogrammed library of home management, educational and entertainment programs that allow the user to utilize the full potential of the microcomputer immediately without first mastering a programming language. The "family" computer is now available in department stores and specialty electronics stores. VideoBrain sells for a suggested retail price of $500 and is FCC approved for use with home color or black and white television sets.

The computer is fully guaranteed and comes with a complete and easily understandable owner's manual. The unit also includes everything necessry to hook up to a TV and run programs — AC adaptor, TV hook-up cord and antenna switch box, two joy sticks and three cartridge programs. For further information contact VideoBrain Computer Co., 150 Wolfe Rd., Sunnyvale, CA 94086, (408) 737-2680.

CIRCLE INQUIRY NO. 156

Siliconix IC Expands Feasibility of Detector, Alarm and On/Off Control Systems

The L911 detector integrated circuit is a versatile component allowing low cost, micropower sensing alarm and detector systems to now be built. The L911 is the latest addition to the Siliconix micropower linear product line. The L911 contains a MOSFET input comparator, a trip point comparator, internal trip point reference, output driver and pulsing output oscillator. The L911 is a monolithic bipolar-PMOS device.

The L911 is simple to use and easily interfaces with a variety of detectors, ON/OFF control systems and monitor alarms. The system can be compactly built, working on a 9V alkaline battery for one year.

The device is ordered as an L911CJ and is in a 14-lead plastic DIP. Pricing is $2.18 (in 100's) and delivery is from stock. For further information contact Siliconix, Inc., 2201 Laurelwood Rd., Santa Clara, CA 95054, (408) 246-8000.

NO CIRCLE INQUIRY NUMBER

A $10 NMOS 16-Bit Microprocessor

Using a newly developed n-channel metal oxide semiconductor process, National Semiconductor Corporation has gone into production on a sophisticated, high throughput 16-bit microprocessor that outperforms most present 8-bit CPU designs.

Designated the INS8900, the 40-pin device has an interrupt structure, addressing modes and logical capabilities traditionally associated with minicomputers.

More importantly, instruction execution times for most of the commonly used routines on the INS8900 are equivalent to those on advanced 8-bit designs like the 8085 and 10 to 30 percent faster than on present generation designs, such as the 2 microsecond 8080.

The INS8900 is software compatible with the PACE microprocessor family and can use the same software development aids, including the PACE Universal Development System. For more information contact National Semiconductor Corp., 2900 Semiconductor Dr., Santa Clara, CA 95051, (408) 737-5956.

CIRCLE INQUIRY NO. 153

Brochure Features 16-Position DIP Switch

A full-color brochrue features EECO's 16-position binary coded MICRO-DIP, "the World's smallest coded dual-in-line switch."

Four pages cover photos, description, features, options, technical specifications and code truth tables for the subminiature fully-coded DIP switch that is designed to be mounted directly on a P.C. board or inserted into a standard IC socket.

For more information contact EECO, 1441 E. Chestnut Ave., Santa Ana, CA 92701, "Switch Products", (714) 835-6000.

CIRCLE INQUIRY NO. 159

Version 3.0 Of Super BASIC

Technical Design Labs of Princeton, New Jersey is introducing Version 3.0 of Super BASIC. Version 3.0, of Super BASIC, provides programmable error handling that allows the user to specify special error handling routines processing any error occurring in the basic program without aborting the program.

Version 3.0 of Super BASIC is being released under CP/M Version first and later as a serial paper tape version. The CP/M Version allows for the dynamic assigning of arbitrary disc files to the reader/punch of standard 3.0 Zapple Super BASIC. The non CP/M version of Super BASIC 3.0 is the only non-disc that allows data file input and output.

Super BASIC Version 3.0 is on a diskette and is part of TDL's Software Package A which consists of Version 3.0, The Macro Assembler 2.2, Z-TEL Text Editing Language and the Text Output Processor. This entire package is available now for $249. For further information contact Technical Design Labs, Research Park, Bldg. H, 1101 State Rd., Princeton, NJ 08540, (609) 921-0321.

CIRCLE INQUIRY NO. 160

MacroFloppy™ by Micropolis

A new series of fully packaged and assembled disc drives priced as low as $695 — including software, S-100 compatible controller and 143,000 byte capacity is available from Micropolis Corporation.

Developed for the retail microcomputer market, the new MacroFloppy™ series initially includes two versions:

Model 1041 packages a drive, enclosure, cabling and connectors, disc operating system and disc extended BASIC at a suggested retail price of $695. It is intended for integration into any 8080A or Z-80 microcomputer chassis.

Model 1042, suggested retail of $795, adds a power supply and DC regulators for desktop use.

Delivery is 30 days on both models, and they are factory warranted for 90 days.

The disc operating system offered with each MacroFloppy subsystem is a complete package including assembler, file management routines and utilities to support 8080A and Z-80 programs. For additional information contact Micropolis Corp., 7959 Deering Ave., Canoga Park, CA 91304, (213) 703-1121.

CIRCLE INQUIRY NO. 157

Tape 'N Text Modular Lessons

Tape 'n Text learning programs consist of audio cassette tapes coordinated with printed texts. The learner is afforded the opportunity to hear on tape what he sees in the printed texts at the same time. In addition the audio tapes provide clarifying information not in the text.

For an additional charge of $2 a hard copy listing of BASIC programs can be ordered for each Tape 'n Text series. The programs are designed to test the learner's knowledge and skills attained at the completion of each Tape 'n Text unit. Computer hobbyists can set up their very own computer based family learning and testing center at home for the study of

CIRCLE INQUIRY NO. 76

PSSST!

FEBRUARY SPECIALS

• lowest prices • immediate delivery • free shipping

IMSAI 80/30 Video Computer

$1099 **(reg. $1199)**

lowest cost complete systems package on the market

A complete computer with memory, built-in monitor (full 80 x 24 display), powerful monitor software, serial and parallel ports, IKB-1 Intelligent Keyboard. Key operated.

- 8085-based for 50% faster program execution (3 mHz) using standard 500ns memory
- 10 slot motherboard, 28-amp power supply
- video display with text editing features
- optional: fully-slotted, wave soldered motherboard, $75

IMSAI Disk Drive with Multiple Controller

$699* **(reg. $770)**

drive over *one Megabyte* of disk storage

Drive up to four standard floppy disks and three mini-floppies *simultaneously.* Single or double density. Program controllable. Includes one mini drive, DIO. Perfect for your new IMSAI 80/30. Shown with optional cabinet.

*add $39 for power supply, $39 for cabinet

IMSAI 80/15 just add love, one tv, and a little memory

$699 **(reg. $799)**

New! Available now at a special introductory price. Simple, easy to use operator's panel. Comes with 10-slot terminated and regulated motherboard. Video capability.

- 8085 processor, 3 mHz clock rate
- parallel & serial ports
- optional: fully-slotted, wave soldered motherboard, $75

$249 **(reg. $275)**

IKB-1 Intelligent Keyboard

- 8048 processor controlled functions
- parallel or RS-232 interface
- user-programmable keys and type faces

Texas Instruments Impact Printer

$1895**

TI reliability, plus high-speed smart printing

TI's new 810 multi-copyserial impact printer. Lets you print up to 440 lines per minute because of its unique look-ahead bi-directional feature, controlled by an on-board microprocessor.

- 150 cps, 110 to 9600 baud switchable
- includes tractor feed
- programmable forms feed available
- RS-232C serial interface

See these products at the following MICROAGE locations:

Phoenix Byte Shop	Tempe Byte Shop	Tucson Byte Shop	Dallas Byte Shop	MicroAge Service Center
24 W. Camelback	813 N. Scottsdale Rd.	2612 E. Broadway	1474 W. Spring Valley Rd.	803 N. Scottsdale Rd.

MICROAGE™ MAIL ORDER
803 N. Scottsdale Rd.
Tempe, AZ 85281
(602) 894-1193

Item	Price	Qty	Amount
IMSAI 80/30	$1099		
IMSAI Disk Drive & Controller	699		
Disk power supply (optional)	39		
Disk cabinet (optional)	39		
IMSAI 80/15	699		
Wave soldered motherboard (optional)	75		
IKB-1 Keyboard	249		
TI 810 Printer (** add $25 shipping)	1895		
40-pg. BYTESHOPPER GUIDE	2.50		
(Az. res. add 4%)	Total		

Check one:
☐ Check (15-day delay for clearance) ☐ Master Charge
☐ Cashier's check ☐ VISA

Card No. ____________________
Interbank No. ____________ Exp. Date ______
(Save time . . . call collect on credit card orders)
Name ____________________
Address ____________________
City __________ State ______ Zip ______
Phone ____________________

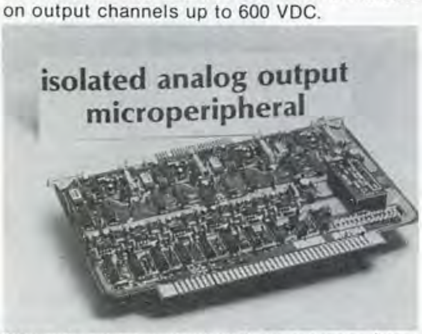
algebra, computer math, English, and BASIC programming language.

The following Tape 'n Text programs are available: Program CA *Programming in BASIC*, Program MA *The Second Language and Contemporary Applications*, Program MB *Basic Math for Computers*, Program MC *Beginning Algebra Course*, Program EA *Basic Language Usage*. Each contains 4 cassette tapes and 4 printed texts, individually packaged, for $19.95. For further information contact Williamsville Publishing Co., Inc. Box 237, Williamsville, New York 14221.

CIRCLE INQUIRY NO. 161

Isolated 4-Channel Analog Output System

Four optically isolated analog outputs are available for Motorola Micromodule and EXORciser microcomputers. Burr-Brown's MP7504 analog output system protects the microcomputer's data and address bus from transients on output channels up to 600 VDC.



Channels are programmed as memory locations and the address block is selectable and can be placed anywhere in memory. A single instruction sets the input of a D/A converter. The output board is shipped to operate in the halt mode and the user can set it in the poliling mode.

The totally compatible MP7504 operates from the microcomputer's +5 and ± 12V supplies. Price is $695 each. For more information contact Burr-Brown, International Airport Industrial Park, Tucson, AZ 85734, (602) 294-1431.

CIRCLE INQUIRY NO. 158

Single-Chip Microcomputer

The 8021 operates on a single +5V power supply as a self-contained microcomputer system. The single chip contains all functions required for digital processing and control, including 8-bit central processor with more than 60 instructions; 64 bytes of read/write data storage in RAM; 1024 bytes of program storage in ROM; clock generator and other system utilities; interval timer/event counter; 21 I/O lines.

Its instruction set is designed for easy, efficient programming. The set provides a large variety of conditional jumps. To conserve memory and maximize throughput, most instructions are single-byte, single-cycle with no instructions longer than two bytes, two cycles. The instruction cycle time is 10 microseconds.

For more information contact Intel Corporation, 3065 Bowers Ave., Santa Clara, CA 95051. (408) 249-8027.

CIRCLE INQUIRY NO. 115

Universal IC Pluggable Packaging Panels

New Universal IC wire-wrap Packaging Panels, SEries PS4108 provide a multiple interface system with the flexibility of four Input/Output channels for segmenting logic functions.

The new panels consist of 30 columns, with 64 terminals per column, on .100-in. centers. The IC socket contacts are low-profile, high-reliability closed-entry, with one- or two-level wire wrappable .025-in. square terminations.

Available with or without 108-pin I/O connectors. Delivery on the new panels runs two to four weeks; prices range from $1.00 to $2.50 per IC position. For further information contact Garry Mfg. Co., 1010 Jersey Ave., New Brunswick, NJ 08902, (201) 545-2424.

CIRCLE INQUIRY NO. 145

New Computer Warehouse Catalog A System Buyer's Guide

The new 1978 Catalog from Computer Warehouse (CW) is packed with data on 10 different microcomputers, cost-effective used peripherals, and descriptions of over 275 computer books now available for immediate delivery. Over 1,500 products are covered — both new and used — from over 170 different vendors.

Highlighting the catalog are detailed descriptions, hints on the best uses, and a complete comparison chart that covers the microcomputers currently available from CW: Southwest Tech 6800, IMSAI 8080, Processor Tech Sol 20, IASIS 8080, Olivetti P6060, Intersil Intercept, Jr., Commodore Kim, RCA Cosmac, Motorola 6800 and National SC/MP.

With over three times the data of its earlier edition, this catalog will serve as a valuable data reference for anyone involved in building microcomputer systems. Computer Music and Art have been added to this edition.

The 1978 catalog is available for $1 to cover 1st class postage and handling from Computer Warehouse, 584 Commonwealth Ave., Boston, MA 02215, (617) 261-1100.

CIRCLE INQUIRY NO. 152

24-Channel Logic Analyzer Plugs into S-100 Bus

The unit, called the DATALYZER, is designed for use in an S-100 type computer system and has the additional capability of self-monitoring the S-100 address and data lines by simply changing operating modes.

Triggering, display, formatting, and operational modes of the Datalyzer are controlled from the users system input device. The trigger word can be up to 16 bits wide with any combination of 1, 0 or X (don't care) states possible. In the internal mode, the trigger word is compared to the S-100 address bus.

The key to the Datalyzer's power is the monitor program resident in 5K bytes of the user's system memory. The monitor provides the operator with the capability of parameter modification, multi-reference memory comparison, control mode selection, and S-100 data assembly. Once the control parameters are set up, the monitor can be exited for later return and subsequent acquisition data analysis.

The price for the Datalyzer in kit form is $495, while the assembled unit is $595. Included are three probe assemblies, a system monitor on paper tape, and a comprehensive applications manual which can be purchased separately for $7.50. Delivery is 4 weeks ARO. For more information contact Databyte, Inc., P.O. Box 14, 7433 Hubbard Ave., Middleton, WI 53562.

CIRCLE INQUIRY NO. 150

SC/MP Motherboard Kit

Expand your SC/MP Kit (or SC/MP Kit with keyboard) into a system capable of utilizing available SC/MP application boards.

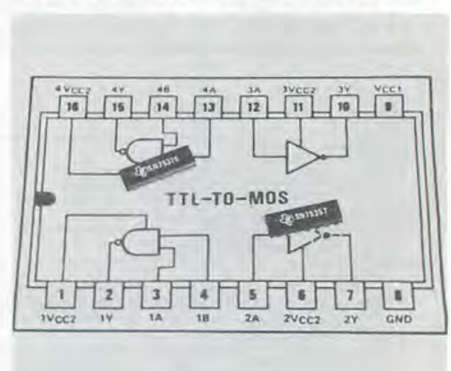
Our Motherboard Kit makes use of your existing monitor and I/O handler (for TTY or Keyboard Kit), and therefore requires memory locations O to FFF to be used for the SC/MP Kit System. The user will then have available memory locations 1000 to FFFF for his application boards. The Motherboard Kit provides one expanded output with provisions to add 72 pin edge connector, making it possible to plug in our 5 plug extender board.

SC/MP Motherboard Kit sells for $99.00 plus $2.00 shipping and handling. For further information contact BW Ward Enterprises, P.O. Box 8122, Long Beach, CA 90808.

CIRCLE INQUIRY NO. 151

Quad TTL-to-MOS Drivers

The SN75357 features three-state outputs; and the SN75375 has individual supply voltages (V_{CC2}) for each of the four drivers, capable of being operated from five to 24 volts.



The SN75357 has very low transient current during switching, making this circuit especially useful for interfacing with CMOS systems.

Individual supply voltage (V_{CC2}) pins on the SN75375 allow individual adjustment of V_{OH} levels to match various load conditions.

Both circuits are characterized for operation from 0° to 70°C. They are offered in either plastic (N suffix) or ceramic (J suffix) dual-in-line packages and are available through TI authorized distributors or from TI in Dallas. For prices and additional information contact Texas Instruments, Inc., P.O. Box 5012, M/S 308 (Attn: SN75357/SN75375), Dallas, TX 75222, (214) 238-2165.

CIRCLE INQUIRY NO. 137

Standard Precision Gear Heads

Featuring ratios up to 17,000:1 and backlash 30-minutes maximum, Harowe Servo Control's Planet Division offers six basic sizes of gear heads — 5, 10, 11, 15 and 18. Respective model numbers are A101A, A102B, A103A, A104A, A105A and A106A.

Material is stainless steel throughout, except black anodized aluminum housing and adapter. Features include low starting torque and temperature-range capability of -55° to 125°C.
For more information contact Harowe, Planet Instrument Div., P.O. Box 547, West Chester, PA 19380, (215) 692-2700.

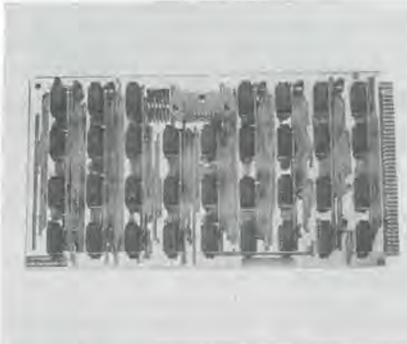
CIRCLE INQUIRY NO. 129

Dual Floppy-Disk Interface

Pacific Cyber/Metrix, Inc. has available a dual floppy-disk interface module for its 12-bit microcomputer system, the PCM-12.

The PCM-12 system is software-compatible with Digital Equipment Corp.'s PDP-8 family of minicomputers.

PCM's floppy-disk module interfaces the PCM-12 to Data Systems Design's Model 210



floppy-disk memory system and sells in single quantities for $259 assembled and $169 in kit form. Volume discounts range from 10 to 25 per cent.

Fully plug-compatible with DSD's Model 210, PCM's 12440 floppy-disk module will allow PCM-12 users to execute all PDP-8 floppy-disk diagnostics and makes the PCM-12 system fully compatible with all mass storage operating systems already developed for the PDP-8 family of minicomputers.

For more information, contact PCM, Inc., 3120 Crow Canyon Rd., San Ramon, CA 95483, (415) 837-5400.

CIRCLE INQUIRY NO. 122

Digital Pulser Probe

The new DO-1 Digital Pulser from Continental Specialties Corporation, in addition to some substantial performance specifications, boasts a tiny bit of automation.

Internal circuitry monitors the node being probed, then presets the dual mirror output circuitry to pulse the node the other way. It delivers a strong enough pulse (50ma in the CMOS mode, 100ma in the TTL mode) to kick most lines with no need to desolder, unplug or isolate. And the output is short circuit protected. It can drive a dead short indefinitely with no danger of damage to the Pulser.

The $74.95 DP-1 Digital Pulser is one member of the new, inexpensive digital trouble-shooting hardware family CSC calls The Logical Force™. For additional information contact Continental Specialties Corp., 44 Kendall St., New Haven, CT 06509, (203) 624-3103.

CIRCLE INQUIRY NO. 124

OCR-A Scanner

A hand-held scanner which automatically reads prices and other information on merchandise tags in department stores has been released for sale by NCR Corporation.

The NCR 7867, a pistol-shaped device weighing only 6 ounces, is moved by the salesperson over the merchandise tag. The information is printed in an Optical Character Recognition (OCR) type font which can be read by people as well as machines.

The scanner reads the data, edits it and transmits it to the NCR retail terminal to which it is attached. The scanner is an option available immediately with the NCR 280 and 250 systems. It will be released for the NCR 255 and 2151 systems in the future.

For more information contact NCR Corporation, Dayton, Ohio 45479, (513) 449-2150.

CIRCLE INQUIRY NO. 149

Low-Cost Bus

S-100 users will receive maximum system flexibility provided by this low cost bus from T & H Engineering. The power supply produces + 8V 8 amps and ± 16V 1 amp. Two of the back plane's 8 slots are spaced for W/W boards. Models LCB-4 and LCB-8 come with 4 and 8 W/W connectors and guides, respectively.

An input voltage control provides cooler operating temperature. Both kits and assembled versions are available: LCB-4 assembled $167.00, kit $149.00. LCB-8 $198.00 assembled, Kit $174.00. Cover (vinyl clad) and end plates (rear with 6 "D" shell mounting holes) $25.00. Dealer inquiry invited. Order from T&H Engineering, P.O. Box 352, Cardiff, CA 92007.

CIRCLE INQUIRY NO. 111

Film Capacitors

New flame retardant high performance film capacitors in several case styles have been added to the H-Series of metalized polycarbonate capacitors available from W-K Industries.

Incorporating an advanced state-of-the-art design, flame retardant versions of the H-Series capacitors feature the same sizes and electrical characteristics as non-flame retardant styles. Flame retardant capacitors are available in wrap-and-fill and epoxy packaging, and also in round and rectangular hermetically sealed styles.

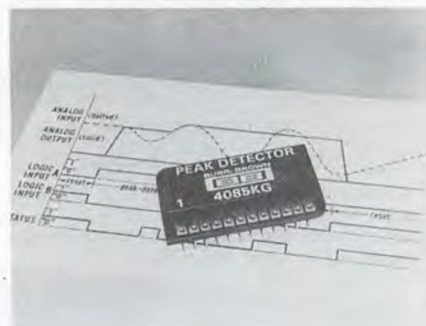
The new capacitors are specifically designed for applications in aircraft and computer systems as well as in industrial and commercial electronic equipment where flame retardance is essential to the safety of people and systems.

For additional information on flame retardant H-Series metalized polycarbonate capacitors contact W-K Industries, 1960 Walker Ave., Monrovia, CA 91016.

CIRCLE INQUIRY NO. 139

IC Peak Detector

Burr-Brown's new 4085 Hybrid Peak Detector is the first circuit of this type to be offered in a hybrid circuit package, and the only one available over the full military temperature range.



This completely self-contained circuit tracks input signals over the range of -10V to + 10V with ± 15V supplies and holds the peak value until reset or until a higher peak is detected.

Three versions are available. The ceramic 4085KG has a specification range of 0 to + 70°C. The 4085BM and 4085SM are hermetically sealed metal packages with specification ranges of -25 to + 85°C and -55 to + 125°C respectively.

Prices for the 4085KG are $49.00 (1-24), $45.00 (25-99) and $36.00 (100-249). The 4085BM is $64.00, $57.00 and $42.00, respectively. The 4085SM is $81.00, $74.00 and $59.00. For more information, contact Burr-Brown, International Airport Industrial Park, Tucson, AZ 85734, (602) 294-1431.

CIRCLE INQUIRY NO. 121

Switching Power Supplies

A new range of switching power supplies combining the high-reliability, size and weight advantages of switching technology with low cost is available from the power supply operation of Gould, Inc. The SMG Series is especially useful for enclosed unit applications in the data processing and telecommunication areas where high-reliability is required.

The SMG Series consists of twelve 5V models at current levels from 1.6A to 450A, three 12V models and three 24V models. All units operate from a 115VAC input.

 CIRCLE INQUIRY NO. 88

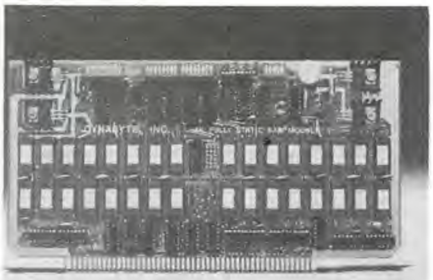
In the event of a power failure, the SMG Series units maintain system operation by providing one full cycle of holdup protection. All 18 units in the SMG Series are currently available from stock and are covered by a two-year warrantee. For information contact Gould, Inc., Electronic Components Div., 4601 N. Arden Dr., El Monte, CA 91731, (213) 442-7755.

CIRCLE INQUIRY NO. 118

16K, 32K Static RAMs

Two fully static RAM modules for the S-100 bus ar now available from Dynabyte, Inc.

The 16K static RAM and the 32K static RAM are available with access times of either 450 nanoseconds or 250 nanoseconds. The 250 nanosecond RAMs are compatible with 4MHz Z-80 processors. Both static RAMs are completely assembled, tested, and burned in and guaranteed for one year.



The 32K static RAM modules offer 4K boundary addressing, complete buffering, and conservative thermal design.

The statis RAM modules employ Schmitt trigger buffers on inputs and tri-state TTL buffers on outputs. For prices and more product information, contact Dynabyte, Inc., 4020 Fabian, Palo Alto, CA 94303, (415) 494-7817.

CIRCLE INQUIRY NO. 134

Time Delay Device

Specifying timers for an application formerly required a specific AC or DC voltage plus a fixed time delay. Artisan's new Model USA (Universal Switch Adjustable) can be used in almost all applications.

This one unique timer will operate in *any* load circuit from 24 to 240 volts AC or DC. This one time delay device can be programmed by the user to delay current to the load from 1 to 1024 seconds. This delay is adjustable by binary code. The new Model USA is completely solid state, encapsulated and will operate faultlessly for millions of cycles. It is UL recognized and CSA approved.

For technical and price information contact Artisan Electronics Corp., 5 Eastmans Rd., Parsippany, NJ 07054. (201) 887-7100.

CIRCLE INQUIRY NO. 142

High-Speed 4K Static RAM

Zilog, Inc., has started production of one of its first two memory products, a 4096-by-1-bit static RAM — the Z-6104 — designed for use with microcomputers and numerous main memory storage applications.

Pin-compatible with Mostek's MK 4104 static RAM, the new Zilog device features access time as fast as 100ns — a result of the company's N/MOS silicon-gate process — and low standby power supply current.

To broaden its use, the Z6104 is available in five speeds — from 100 nanoseconds to 300ns access time. The 100ns version (Z-6104-1) is for large-scale mainframe applications where computer speed is a prime consideration. The 200ns (Z-6104-3), 250ns (Z-6104-4) and 300ns (Z-6104-5) versions are well suited for a wide range of microprocessor applications.

Pricing for 100-999 quantities in ceramic packages is $20 for the 250ns circuit and $22 for the 200ns version. The 4K static RAMs are available off-the-shelf from Zilog distributors. For more information, contact Zilog, Inc., 10460 Bubb Rd., Cupertino, CA 95014, (408) 446-4666.

CIRCLE INQUIRY NO. 133

Microprocessor-based Mini-floppy

The Digidisk™, a microprocessor-based mini-floppy disk drive that emulates paper tape and magnetic tape, allows the operator to prepare and edit messages and data off-line and then transmit them through the optional built-in acoustic coupler. The unit will operate with any data communication printer or CRT terminal through the standard RS-232 interface.

The Digidisk™, which responds to remote control code from a computer, is ideal for data storage, automatic send/receive program loading, and customer-oriented data collection applications. The portable, lightweight unit uses the new flexible "mini-disk" with a capacity of over 100,000 characters.

Prices for the Digidisk™ with standard RS-232 interface are $1199; $1299 with built-in 300-baud originate-only coupler; $1699 with a built-in 1200-baud 202-compatible coupler. For further information contact Digicom Data Products, Inc., 1440 Koll Circle, Suite 108, San Jose, CA 95112, (408) 279-8711.

CIRCLE INQUIRY NO. 120

APF Introduces PeCos One

The PeCos One is a personal computer that can perform all the functions that were heretofore available in computers costing tens of thousands of dollars. PeCos represents both a price breakthrough and a design achievement because, with the PeCon One, anyone can learn to use and program a personal computer.

APF is launching a new breed of computers which talk to the user in a non-technical, easily-mastered language — English.

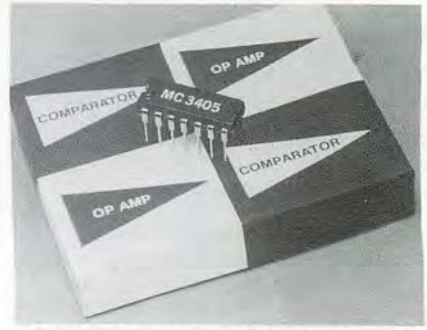
PeCos One can already be expanded to meet just about any personal use, with currently available options including two additional cassette units and 24K of memory. In the future APF will also offer an add-on printer.

To obtain further information contact APF Electronics, Inc., 444 Madison Ave., New York, NY 10022, (212) 758-7550.

CIRCLE INQUIRY NO. 155

Two of A Kind

A versatile new IC, Motorola's MC3405/3505, combines two different functions: a pair of operational amplifiers, similar to type MC3403/3503, and a pair of DC comparators, similar to type LM339/139.



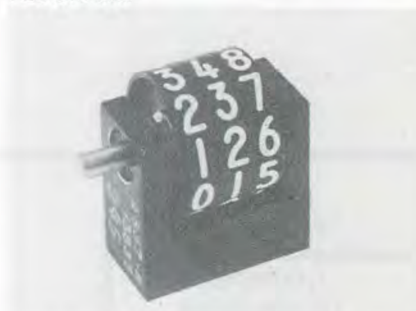
The combination is useful in linear subsystems in which the internally compensated, linear input and output characteristics of an op-amp, and the switching speed and logic-compatible output of a comparator are required in adjacent circuit areas.

Available from stock, the MC3405/3505 is available in two temperature ranges in plastic (P) and Ceramic (L) 14-pin DIL packages. The MC3405 has a specified operating ambient temperature range of 0°C to +70°C, while the MC3505 is specified from -55°C to +125°C. For prices and further information contact Motorola Semiconductor Products, Inc., P.O. Box 20912, Phoenix, AZ 85036, (602) 962-3151.

CIRCLE INQUIRY NO. 130

Three-Digit Decimal Counter

Up to 350rpm input speed with Harowe's Planet Division's three-digit decimal counter, Model A-837-2. Features include .672 drum diameter with .187 high numerals per MS-33558 dull white on dull black background; colors per FED-STD-595.



Unit cycles 000 through 999 and returns to 000, continuous and reversible at any point. Drive-direct to first drum, one revolution of input shaft provides increase of 10 units of indication.

For more information contact Harowe, Planet Instrument Div., P.O. Box 547, West Chester, PA 19380.

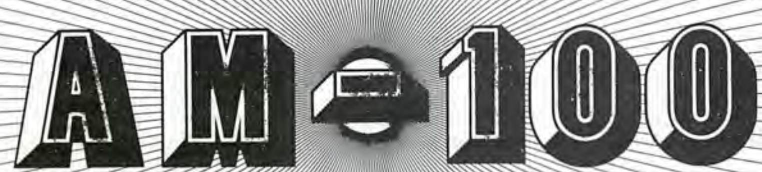
CIRCLE INQUIRY NO. 110

MIGET

A new, very-low-cost, flexible data communications terminal called the MIGET (Miniature Interface General-purpose Economy Terminal) provides keyboard entry and display output compatible with all microcomputer and microprocessor systems using RS-232C interface and ASCII code.

WE'RE TAKING ANOTHER BYTE!

Byte Shop of Pasadena is now doubling its current facilities to meet your computing needs quicker and more efficiently than other stores in the Greater L.A. Area. For example, we are tripling our educational and library facilities. You will also find our service department to be exceptional, as well as a variety of dependable systems up and running. We're only a few blocks south of the 210 (Foothill) Freeway on Lake Avenue. Stop by and see what a full service computer store is like!



NOW AT

You have to SEE it to BELIEVE it!

The Alpha Microsystems AM-100 is LIGHT YEARS ahead of everything else you've seen so far in the low cost computing field.

For a FRACTION of what you'd normally pay for the SOFTWARE ALONE, you get a 16-bit processor with ALL of these BIG-SYSTEM capabilities:

MULTI-TASKING, MULTI-USER TIMESHARING

- DEVICE INDEPENDENT I/O
- ADVANCED FILE STRUCTURE
- POWERFUL SYSTEM COMMANDS
- SOPHISTICATED TEXT EDITOR
- FULL MACRO ASSEMBLER
- LINE PRINTER SPOOLER
- RE-ENTRANT, MULTI-USER BASIC *COMPILER*
- LARGE UTILITIES LIBRARY

Yet, with all this it's still compatible with the S-100 BUS!

If you like the Decsystem-10 operating system, if you like TECO . . . if you like the PDP-11 instruction set . . . you'll LOVE the AM-100!

ONLY
***$1495**
IN STOCK *NOW!*

BYTE SHOP
of Pasadena

496 S. LAKE AVE.
PASADENA, CA. 91101
PHONE: (213) 684-3311

HOURS: Tuesday-Friday, 12:00-8:00
Saturday, 12:00-6:00
Closed Sundays and Mondays

master charge — BANKAMERICARD

SYSTEMS BASED ON:
Alpha Micro System AM 100
Poly 8813
Industrial Microsystems
Cromemco Z2D System 1
Processor Technology Sol 20
Imsai 8080, 8085 Systems
Vector Graphic
TEI
Parasitic Engineering

DISC SYSTEMS
PerSci
Calcomp Trident 50mb-900mb
Shugart
iCOM
Micropolis
North Star
Industrial Microsystems
Cromemco

PRINTERS AND TELEPRINTERS
Texas Instruments Model 810 & Silent 700
Teletype Model 40 & 43
Centronics Model 761 & 799
Diablo 1620 & 1610

CRT TERMINALS
Lear Siegler ADM 3A
Hazeltine 1500
Adds Regent 100 & 200
Textronics Graphic & Alphanumeric

DISPLAY MONITORS
Hitachi
Sanyo
Koyo

KEYBOARD
Datanetics

INTERFACES
Heuristics Speech Lab
Canada Systems Power Control
Tarbell Cassette & Disc Interface
Oliver Audio Eng.
Dynabyte Video Interface
Processor Tech VDM-1, 3P + S

MEMORY BOARDS
Industrial Microsystems 8K, 16K
Dynabyte
Processor Tech 16K
Extensys 64K
Ithaca Audio 8K Bare Boards

SOFTWARE
Microsoft Basic 4.4; Fortran
Digital Research CP/M & MAC
Structured Systems QSort General Ledger
Shrayer: Electric Pencil
BSL Vol. I-VII
Osborne Basic Prog. Discs
CP/M User's Group Collection

Micon Industries has put a price tag of $400 on the MIGET. Even lower prices are available on large OEM quantity orders.

Basic operation of the MIGET is through a simplified typewriter keyboard which provides a simultaneous 32-character display above. The keyboard layout is considerably easier to operate than most higher priced computer terminals and is silent in comparison.

For more information, contact Micon Industries, 252 Oak St., Oakland, CA 94607, (415) 763-6033.

CIRCLE INQUIRY NO. 126

The Box

The Box, a portable computer enclosure manufactured by 2005AD, Inc., is bound to revolutionize the hobbyist computer and has direct applications in industry and business as well. Engineers building dedicated systems from various manufacturers' application cards will find The Box an admirable way to integrate their system into a single sturdy attractive portable package; business men will admire the 'business machine' styling, while hobbyists will appreciate the adaptability of the system.

The Box provides a rugged portable enclosure for anyone needing to use a microprocessor in the field, for anyone who works both at home and in the office, and for anyone needing a good-looking, adaptable enclosure which is easily modified without extensive shopwork. Prices start at $299 for The Box, (either portable or rack-mount style), main chassis and keyboard metalwork, with all hardware. A power supply kit, fan kit and S100 wire wrapped frames are available for an extra charge.

For more information contact 2005AD, Inc., 2005 Naudain St., Philadelphia, PA 19146, (215) 545-3488.

CIRCLE INQUIRY NO. 147

Ruggedized Punched Tape Reader Catalog

A 20-page catalog covers EECO's very broad line of ruggedized punched tape readers for airborne, shipboard, mobile van, flight line and military applications.

The contents include information about tape punches, tape, tape reels, interfaces, conformance to MIL specifications and support services.

For your copy of EECO's catalog ofRuggedized Punched Tape Products, contact EECO, 1441 E. Chestnut Ave., Santa Ana, CA 92701, (714) 835-6000 "Tape Reader Products".

CIRCLE INQUIRY NO. 125

1978 Radio Shack Catalog

The new 1978 Radio Shack Catalog, the company's 30th consecutive issue, is now available from Radio Shack stores and dealers nationwide.

The 164-page catalog includes 100 full-color pages describing the company's exclusive line of products for home entertainment, hobbyists, CBers and experimenters.

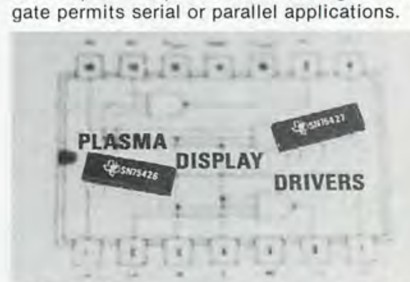
An insert card in the catalog introduces Radio Shack's new TRS-80 Microcomputer System. The new catalog also lists hundreds of specialized electronic items, parts and accessories, tools, tubes, semiconductors, wire and cable, intercoms, microphones, timers, batteries and a complete library of Radio Shack's own books on electronics and related subjects.

Radio Shack's 1978 Catalog #289 is available free on request from all Radio Shack stores and dealers. For further information contact Radio Shack, H.L. Siegel, 2617 W. 7th St., Ft. Worth, TX 76107, (817) 390-3272.

CIRCLE INQUIRY NO. 148

Quad Plasma Display Drivers

Two integrated circuit AC plasma display drivers, the SN75426 and the SN75427 feature 90-volt output swing, CMOS compatible inputs, high data input impedance, typically one megohm, and 30 milliampere clamp diodes on the output. Independent addressing of each gate permits serial or parallel applications.



Logic of the two drivers is complementary to permit controlled writing or erasing at a specified point on the display. The output of the SN75426 non-inverting pulser is normally near ground potential and is pulsed to near Vcc2. The output of the SN75427 inverting pulser is normally near Vcc2 potential and is pulsed to near ground potential.

The devices require two power supplies, one for the logic section and one for the high-voltage outputs. Both drivers are characterized for operation from 0 to 70°C. The parts are priced, in 100-piece quantities, at $1.41 each in plastic packages and $1.76 in ceramic.

For more information contact Texas Instruments, Inc., Inquiry Answering Service, P.O. Box 5012, M/S 308 (Attn: Plasma Display Drivers), Dallas, TX 75222, (214) 238-2165.

CIRCLE INQUIRY NO. 1638

TI Second Sources MC3446

Texas Instruments announces it has second sourced the MC3446 quadruple bus transceiver IC for IEEE-488 applications. This quadruple,

MAY THE FORCE BE WITH YOU

See the P.E.T. 2001 Computer from Commodore at

THE COMPUTER STORE
820 Broadway

Santa Monica, California 90401
Phone (213) 451-0713

THE MOST EXPERIENCED COMPUTER DEALERS ANYWHERE

Store Hours: Tues — Fri Noon — 8 pm • Saturday 10 am — 6 pm

Located 2 blocks North of
the Santa Monica Freeway
at the Lincoln Blvd. exit

Phone and mail orders invited
BankAmericard/Visa and
Master Charge welcome.

Authorized dealer for MITS, Commodore, Apple, Cromemco, Micropolis, MECA, Equinox, Microterm, Extensys, SOROC, Micro Computer Devices, Multiterm, Binary Systems, Micro-Ware, Sanyo and Hitachi.

single-ended transceiver is designed for bidirectional flow of data.

The bus terminal characteristic complies with paragraph 3.5.3 of IEEE Standard 488. Each driver output is tied to the junction of an internal voltage divider that sets the no-load output voltage and provides bus termination.

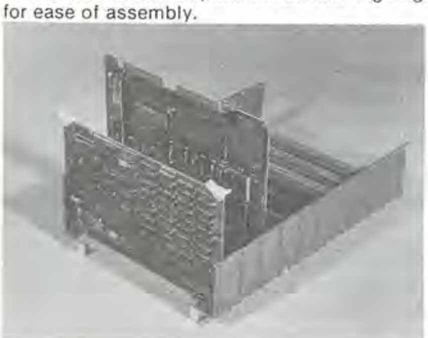
Driver outputs are guaranteed to be "off" during power up and power down if either input is high. The receivers feature 950 millivolt typical hysteresis for noise immunity.

The MC3446 is characterized for operation over the 0° to 70° temperature range. It is offered in either a 16-pin plastic (N suffix) or ceramic (J suffix) dual-in-line package. In 100-piece quantities, the MC3446N is $2.12 and the MC3446J is $2.61. For more information contact Texas Instruments, Inc., IAS, P.O. Box 5012, M/S 308, (Attn: MC3446), Dallas, TX 75222, (214) 238-3527.

CIRCLE INQUIRY NO. 141

Card Cages

Audio Engineering introduces a card cage for use with the MEK6800D2 kit from Motorola, and other compatible computer systems. The unique design incorporates molded plastic card guides, aluminum socket supports, and a mother board. All components are self aligning for ease of assembly.



The card cage is available in a 5-slot and an 11-slot version. Spacing between positions is 1⅜ inches. This allows wire wrap sockets to be used at all board locations.

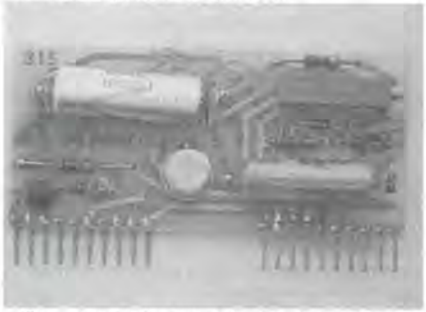
The fully assembled 5-slot card cage is $84.95 and the 11-slot version is $157.50. Both units are available unassembled for $69.95 and $137.50 respectively. Availability is from stock.

A card cage can be designed to customer specifications for other systems. For further information contact Audio Engineering, 121 Wisconsin N.E., Albuquerque, NM 87108, (505) 255-6451.

CIRCLE INQUIRY NO. 135

Integrator/Accumulator

International Microtronics Corporation has added an Integrator/Accumulator to its line of options for the Series 300 Digital Panel meter. The option is designed for applications requiring measurement of current where time is involved as a part of the process.



The standard slew rate for the option is 0.1V/sec., with a drift vs. time ratio of less than 1 microvolt/sec., and ramp linearity of ±0.1%. The unit also has the capability of enabling or disabling the integrator for use in interval to voltage measurement.

The price for the Integrator/Accumulator option in quantities of 1-9 is $48. Delivery is stock to three weeks. For more information contact International Microtronics Corp., 4016 E. Tennessee St., Tucson, AZ 85714, (602) 748-7900.

CIRCLE INQUIRY NO. 132

New from Echo

A complete line of polyester films (mylar), suited for the majority of your graphic art, drafting or computerized plotting needs. Twelve sizes of cut sheets in stock. Buy one sheet or take advantage of our 100 sheet per size discount. Roll stock available in eight widths. Choose from two registration punches: Graphic punch-3/16" slot, 1/4" round, 3/16" slot on tenth inch centers. Drafting Punch-1/4" slot, 1/4" round-maximum spacing 30 inches. Aluminum registration pins with poly tabs available for either type of punch at $1.00 per pin.

For more information, price sheets and samples contact Echo Design & Development Corp., 195 E. Gish Rd., San Jose, CA 95112, (408) 292-0918.

CIRCLE INQUIRY NO. 143

Power Supply Applications Guide

Standard Power, Incorporated announces the publication of a new 10-page technical bulletin entitled, "Power Supply Applications Guide."

Developed by the Engineering Department, Standard Power, the bulletin provides technical information on various applications, enumerating power requirements, e.g., specific designations in voltage, current, overvoltage protection and other electrical parameters.

Applications such as Bubble Memory, Panaplex, Floppy-Disc Memories, I^2L, T^2L, minicomputers, microprocessors, operational amplifiers, and CMOS logic are discussed.

An application of power needs and various recommendations of how to satisfy that specific application's input power requirements are included.

The engineering publication is designated Technical Bulletin No. 109 and is available on

CIRCLE INQUIRY NO. 70

request from Standard Power, Inc., 1400 So. Village Way, Santa Ana, CA 92705, (714) 558-8512.

CIRCLE INQUIRY NO. 128

Three New Models in DE/200 Series

The intelligent, random access displays, DE/210 (10-character positions), DE/220 (20-character positions), and DE/232 (32-character positions) all feature an on-board microprocessor which incorporates a character generator, display buffer and refresh logic.

Display evaluation kits (shipped with each initial display module ordered) include a data cable, power connector, and complete documentation. Mounting kits complete with bezel, filter, and hardware are optional. Power supplies are available with either +5VDC or 110/220 VAC input. In quantities of a hundred the displays are priced at $99, $140 and $199, respectively for the DE/210, DE/220 and DE/232.

For information contact Digital Electronics Corp., 415 Peterson St., Oakland, CA 94601, (415) 532-2920.

CIRCLE INQUIRY NO. 117

8K ROMs Rated at 55ns

Monolithic Memories, Inc. has a series of 1024-by-8-bit bipolar ROMs rated at 55 nanoseconds T_{AA} and priced at $10 in volume for commercial grades.

The new 8K ROMs are also available in MIL-temperature range rated at 70ns access time.

Typical delivery is five weeks from receipt of customer's bit pattern; first-time masking charge is $750 per pattern in minimum quantities of 100.

Monolithic's 8K ROM family comes in a 20-pin package (6289-2), a 22-pin version (6286-2) and a 24-pin package (6280-2) — all designed as drop-in replacements for other vendors' memories. The 22- and 24-pin DIPs are also compatible with Monolithic's off-the-shelf bipolar PROMs.

For higher density requirements, Monolithic also offers 9K (6260-1) and 10K (6255-1) ROMs, both at 100ns speed; and its 16K ROM (6275-1), the industry's fastest at 110ns.

For more information contact Monolithic Memories, 1165 East Arques Ave., Sunnyvale, CA 94086, (408) 739-3535, ext. 111.

CIRCLE INQUIRY NO. 114

System 29

The industry's first development system for prototyping and programming microprogrammed systems, regardless of their architecture, is System 29, a complete prototyping set-up that can be used to develop hardware and firmware designs, and then can check out the prototype design. This microprocessor-based system includes a CRT console and a dual drive floppy disc configured and programmed to support the prototyping of microprogrammed designs.

One of the most flexible aspects of this new system is that it comes complete with a full disc operating system, AMDOS™/29. This disc operating system provides rapid access to programs through a comprehensive file management package. The file subsystem supports a named filed structure, allowing dynamic allocation of a file space as well as sequential and random file access.

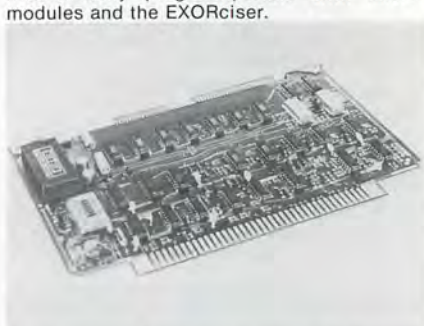
This system, which sells for $24,950, is available from Advanced Micro Devices and its distributors. For more information contact Advanced Micro Systems, 901 Thompson Pl., Sunnyvale, CA 94086, (408) 732-2400.

CIRCLE INQUIRY NO. 146

8-Bit Analog I/O System

Burr-Brown's new MP7400 Analog I/O System is the first 8-bit plug-compatible system offered for Motorola's Micromodules Monoboard Microcomputer and EXORciser® development system, and is the lowest priced analog I/O available for those microprocessors.

The MP7400 contains both analog input and output on a single board and is electrically and mechanically plug-compatible with Micromodules and the EXORciser.



A basic input-only board with eight channels differential or 16 channels single-ended is priced at $295 (1-9) and $198 (100's). A fully loaded board with 32 channels differential or 64 channels single-ended plus two output channels and DC/DC converter is priced at $595 (1-9) and $398 (100's). Delivery for small quantities is two weeks.

For more information, contact Burr-Brown, International Airport Industrial Park, Tucson, AZ 85734, (602) 294-1431.

CIRCLE INQUIRY NO. 144

 CIRCLE INQUIRY NO. 87

BOOK REVIEWS

AN INTRODUCTION TO MICROCOMPUTERS VOLUME 0, THE BEGINNER'S BOOK

Adam Osborne. 226 pages, $7.50
1977 Osborne & Associates, Inc.
Reviewed by Audrey E. Heyser

The introduction of this book states that the book was written for two audiences. For those who have a real interest in learning how to use computers and for those who must live with computers and had better know a little about them. It's the reviewer's thought that this book may tell the second audience somewhat more than they wish to know.

For the first audience the book is well organized, reminiscent of language manuals used by programmers. This book is useful for reference as well as for study and might well be used in a beginning computer classroom. It is interesting to note that INTERFACE AGE Magazine was given credit for the introduction commercially of Floppy ROM™, a read only memory on disc — a well done comment.

Of help to the beginner is that the author uses many drawings and pictures to illustrate concepts often difficult to understand with only words to guide them. Almost every page has some picture frequently combined with comments in the picture itself, giving even greater clarity. In spite of the techniques used to increase understandability this is not a book to be read in two hours.

In the introduction much is said of the possibilities of computer fraud in today's society, yet it was disappointing to find nothing on this subject in the book itself. For those who work only in the general area of computers, for those who don't care if it's an 8-bit machine or not, it would have been immensely valuable to have mentioned a few small, but vitally important things that can be done to minimize fraud, for example, separation of duties, programming standards, counts of records, standard auditing procedures, etc.

The book seems structured for the new mini or microcomputer user. It will make a good basis for an individual interested in computer hardware and perhaps software to a lessor extent, but for the second audience referred to earlier, it is not very valuable. It is too technical to stimulate interest from a tired supervisor. It gives little information to help him to use the computer in increasing company efficiency or to understand problems which frequently develop in the computer room or to work better with computer personnel. I am thinking of the problems caused by the cold in the computer room, by programs which worked well for eight years and suddenly won't work at all, of head crashes and all the other things that can drive otherwise intelligent, sensitive people to be grossly impolite.

I would recommend this book to the members of the first audience. Difficult concepts are made easy and it is well set up for reference use because of a quick index which follows the table of contents.

THE UNIVERSAL ELIXIR, AND OTHER COMPUTING PROJECTS WHICH FAILED

By Robert Glass. 88 pages, $7.50

Reviewed by Audrey E. Heyser

Robert Glass is somewhat like the farmer who hit his plow mule over the head with a "2x4" to get his attention so he could make the mule pull the plow. Mr. Glass gets the attention of his readers in the introduction with his great sense of humor and unique approach so he can teach us something. Rather than use a positive approach of describing computer philosophy, procedures, policies, etc. to teach us the best way to do things, he teaches us by the negative method. The book consists of short stories of disasters centered around computers. Each short story, though dull, starts out well enough, but seems to fizzle out at the end. Thumbing through the book one finds many funny drawings which have only a slight connection with the stories.

The introduction was so funny and thoughtful that it was a kind of promise to the reader of the pleasure and vicarious experience he would gain by reading the short stories. But the promise is not fulfilled.

The one redeeming quality may be found at the end. A chapter called How to Succeed at Admitting Failure. A warm, sensitive and honest appraisal on how to handle failure. It is a chapter well worth reading.

SOFTWARE EDITORIAL

By A. A. Perez, Software Editor

Over the past several months we have printed many types of software articles ranging from the extremely complex business programs by Bud Shamburger, to eloquently written small games like Gary Young's Tax Man. During this period of time we have been reasonably fortunate in determining the reliability of the programs.

Reliability of software is important to us as a magazine. We want the reader to view us as a use book, something that can be picked up with a feeling of what is presented is worthwhile. Occasionally a problem arises, and is not only addressed by you the reader, but by the editorial staff.

To fully appreciate what happens when a software problem arises, it's important to view the process of software evaluation from the magazine's point of view.

Rather than bore you with the mundane tasks of article acceptance, take for granted that we are discussing software that has been accepted, and is being prepared to appear in some issue of the magazine.

Basically software can be divided into five broad categories: The first development software covers the areas of assemblers, algorithms, loaders, and operating systems. This is not to say that all development software falls within this narrow frame, but they are representative. The next category is tutorial software, this type of software can be an actual working program, or a discussion of a concept. In some cases it is a mixture of both. The last three categories: games, business, and scientific software can be grouped together, due to the fact that they all require legitimate working programs to be worthwhile.

Therefore, the first order of business for the software editor, or evaluator is to determine what basic category the software falls under. Armed with this information the task of determining the viability of the software becomes of utmost importance. All five categories present a different set of problems and require a definite method of evaluation.

Tutorial software requires that the concept being presented be fully researched. Not only through the author, but by resorting to the time honored method of searching the literature. Errors and misinterpretations are then edited out, and the final article is put together for publication.

Software that presents a working program of some sort must also be researched, and of course tried, to insure its reliability. At INTERFACE AGE we are able to test, inhouse, most BASIC programs, and many 8080 and 6800 assembly language programs. For software that we are unable to test correctly, or do not understand, we forward it to one of several evaluators. These evaluators, in our opinion, possess the knowledge and expertise to make the necessary corrections and comments.

However, even with all these so-called safeguards built in, something slips by to make us look a little bit stupid. Our saving grace is the reader who quickly rises to the need and pinpoints the problem.

I would like to say that we purposely put in a zinger to see if you're awake, but that is definitely not the truth. Our goal is to provide the best possible software available.

Throughout the course of 1978 you will be seeing more Floppy ROMs™, useful development software, games, and general software. The main thrust will be small business software. Our December issue will tie it all together with a complete book of business programs and a double-sided Floppy ROM.

Hopefully this will give you an idea of the process we go through month after month. We wish that the system was perfect, but it isn't and consequently we are open to any suggestions you may have.

Turning now to what's available this month. The January issue presented the first part of Gary Young's Tax Man program. We now give you the full, complete listing of the game and, yes, it does run. The second article presents an old method of error correction and Ed Comer gets the point across in "Hamming's Error Corrections."

Alan Miller graces our pages once again with another piece of well-written software in "TAPEMON." Following right behind is Fred LaPlante's article on Random files, possibly the best one on this subject to date.

The last two articles shift gears a little by providing something for the calculator buffs, and game players. Professor Martin Patt discusses a method of using your programmable calculator to model experimental data. Professor Patt not only presents his subject in detail, but gets you to wondering about other applications for programmable calculators. Joe Jaworski puts the cap on this month's software section by letting you rule the world, in his article "World Power."

This month's software articles should keep you all busy until next month.

The Tax Man — Part 2

By Gary O. Young

For those of you who enjoyed reading Gary's Tax Man article, in the January issue, here is the complete game listing. We think that once you get the program up and running you will find it not only fun but informative.

—Editor

```
10000 THE TAX MAN (1040 PROGRAM)
10100 REM BY GARY YOUNG
10200 REM      PO BOX 1532
10300 REM      STUDIO CITY, CA 91604
10400 REM WRITTEN FOR AN IMSAI 8080 USING NORTHSTAR DOS BASIC AND
10500 REM 24K OF MEMORY
10600 REM JULY 1977
10700 REM
10800 PRINT "1040 TAX PROGRAM VERSION 4"
10900 DIM T1(4,18),T2(4,11)
11000 DATA 0,2200,0,0
11100 DATA 2200,2700,14
11200 DATA 2700,3200,70,15
11300 DATA 3200,3700,145,16
11400 DATA 3700,4200,225,17
11500 DATA 4200,6200,310,19
11600 DATA 6200,8200,690,21
11700 DATA 8200,10200,1110,24
11800 DATA 10200,12200,1590,25
11900 DATA 12200,14200,2090,27
12000 DATA 14200,16200,2630,29
12100 DATA 16200,18200,3210,31
12200 DATA 18200,20200,3830,34
12300 DATA 20200,22200,4510,36
12400 DATA 22200,24200,5230,38
12500 DATA 24200,28200,5990,40
12600 DATA 28200,34200,7590,45
12700 DATA 34200,40200,10290,50
12800 REM CALIF STATE SINGLE TAXPAYERS
12900 DATA 0,2000,0,1
13000 DATA 2000,3500,20,2
13100 DATA 3500,5000,50,3
13200 DATA 5000,6500,95,4
13300 DATA 6500,8000,155,5
13400 DATA 8000,9500,230,6
13500 DATA 9500,11000,320,7
13600 DATA 11000,12500,425,8
13700 DATA 12500,14000,545,9
13800 DATA 14000,15500,680,10
13900 DATA 15500,1000000,830,11
14000 FOR J = 1 TO 18
14100 READ T1(1,J),T1(2,J),T1(3,J),T1(4,J)
14200 NEXT J
14300 FOR J = 1 TO 11
14400 READ T2(1,J),T2(2,J),T2(3,J),T2(4,J)
14500 NEXT J
14600 DIM A4$(8)
14700 DIM A(60)
14800 K1 = 60
14900 REM M1 IS THE MODE, M2 IS FIRST TAB, M3 IS SECOND TAB
15000 REM M4 = SET PAPER SWITCH
15100 M4 = 0
15200 M2 = 25   M3 = 33
15300 INPUT "NEW OR OLD RUN? ",A1$
15400 IF A1$ = "NEW" THEN 16300
15500 IF A1$< >"OLD" THEN 15300
15600 INPUT "FILE NAME?",A2$
15700 OPEN #0,A2$
15800 FOR J = 1 TO K1
15900 READ #0,A(J)
16000 NEXT J
16100 CLOSE #0
16200 GOTO 16700
16300 FOR J = 1 TO K1
16400 A(J) = 0
16500 NEXT J
16600 GOTO 17300
16700 INPUT "LIST OR UPDATE? ",A3$
16800 IF A3$ = "LIST" THEN 17900
16900 IF A3$ = "UPDATE" THEN 17100
17000 GOTO 16700
17100 IF A1$ = "OLD" THEN 17600
17200 REM M1 = 0 FOR UPDATING NEW
17300 M1 = 0
17400 GOTO 18300
17500 REM M1 = 1 FOR UPDATING OLD
17600 M1 = 1
17700 GOTO 18300
17800 REM M1 = 2 FOR LISTING ONLY
17900 M1 = 2
18000 IF M4>0 THEN 18300
18100 M4 = 1
18200 INPUT "SET PAPER AND RETURN",A3$
18300 J1 = 0
18400 REM UPDATE NEW OR LIST (SEQUENTIAL OPERATION)
18500 IF M1 = 1 THEN 18800
18600 GOTO 19000
18700 REM UPDATE OLD LOOP
18800 INPUT "LINE?",J1
18900 IF J1 = 999 THEN 47200
19000 REM BEGIN CYCLING THRU THE LINES
19100 IF M1< >1 THEN 19300
19200 IF J1< >1 THEN 19800
19300 PRINT "001 NUMBER OF EXEMPTIONS",TAB(M2),%3I,A(8)
19400 IF M1 = 2 THEN 19900
19500 INPUT A(8)
19600 IF M1 = 1 THEN 18800
19700 GOTO 19900
19800 IF J1< >10 THEN 20400
19900 PRINT "010 WAGES, SALARIES, TIPS ",TAB(M3),%9F2,A(1)
20000 IF M1 = 2 THEN 20500
20100 INPUT A(1)
20200 IF M1 = 1 THEN 18800
20300 GOTO 20500
20400 IF J1< >12 THEN 21000
20500 PRINT "012 DIVIDENDS",TAB(M3),%9F2,A(2)
20600 IF M1 = 2 THEN 21100
20700 INPUT A(2)
20800 IF M1 = 1 THEN 18800
20900 GOTO 21100
21000 IF J1< >14 THEN 21600
21100 PRINT "014 INTEREST INCOME",TAB(M3),%9F2,A(3)
21200 IF M1 = 2 THEN 21700
21300 INPUT A(3)
21400 IF M1 = 1 THEN 18800
21500 GOTO 21700
21600 IF J1< >15 THEN 22500
21700 PRINT "015 OTHER INCOME",TAB(M3),%9F2,A(4)
21800 IF M1 = 2 THEN 22600
21900 INPUT A(4)
22000 IF M1 = 1 THEN 18800
22100 INPUT "DO YOU WANT TO DO BUSINESS CALC?",A4$
22200 IF A4$ = "N" THEN 28600
22300 IF A4$< >"Y" THEN 22100
22400 GOTO 22600
22500 IF J1< >40 THEN 23200
22600 IF A(18) = 0 AND M1 = 2 THEN 28600
22700 PRINT "040 GROSS INCOME ON BUS.",TAB(M2),%9F2,A(18)
22800 IF M1 = 2 THEN 23300
22900 INPUT A(18)
23000 IF M1 = 1 THEN 18800
23100 GOTO 23300
23200 IF J1< >100 THEN 24100
23300 PRINT "100 MILES TRAVELED",TAB(M2),%9F2,A(17)
23400 IF M1 = 2 THEN 23700
23500 INPUT A(17)
23600 IF M1 = 1 THEN 18800
23700 C1 = A(17)*.15
23800 PRINT "*** MILEAGE EXPENSE",TAB(M2),%9F2,C1
23900 IF M1 = 2 THEN 24200
24000 GOTO 24200
24100 IF J1< >110 THEN 24700
24200 PRINT "110 PLANE, BUS, TAXI",TAB(M2),%9F2,A(19)
24300 IF M1 = 2 THEN 24800
24400 INPUT A(19)
24500 IF M1 = 1 THEN 18800
24600 GOTO 24800
24700 IF J1< >120 THEN 25300
24800 PRINT "120 ADVERTISING        ",TAB(M2),%9F2,A(20)
24900 IF M1 = 2 THEN 25400
25000 INPUT A(20)
25100 IF M1 = 1 THEN 18800
25200 GOTO 25400
25300 IF J1< >130 THEN 25900
25400 PRINT "130 OFFICE EXPENSES",TAB(M2),%9F2,A(21)
25500 IF M1 = 2 THEN 26000
25600 INPUT A(21)
25700 IF M1 = 1 THEN 18800
25800 GOTO 26000
25900 IF J1< >140 THEN 26500
26000 PRINT "140 DEPRECIATION",TAB(M2),%9F2,A(22)
26100 IF M1 = 2 THEN 26600
26200 INPUT A(22)
26300 IF M1 = 1 THEN 18800
26400 GOTO 26600
26500 IF J1< >145 THEN 27100
26600 PRINT "145 REPAIRS,TAB(M2),%9F2,A(22)
26700 IF M1 = 2 THEN 27200
26800 INPUT A(23)
26900 IF M1 = 1 THEN 18800
27000 GOTO 27200
27100 IF J1< >150 THEN 27700
27200 PRINT "150 TAX ON EQUIPMENT",TAB(M2),%9F2,A(24)
27300 IF M1 = 2 THEN 27800
27400 INPUT A(24)
27500 IF M1 = 1 THEN 18800
27600 GOTO 27800
27700 IF J1< >160 THEN 28900
27800 PRINT "160 JOURNALS AND MAG",TAB(M2),%9F2,A(25)
27900 IF M1 = 2 THEN 28200
28000 INPUT A(25)
28100 IF M1 = 1 THEN 18800
28200 A(35) = A(17)*.15 + A(19) + A(20) + A(21) + A(22) + A(23) + A(24) + A(25)
28300 PRINT "*** TOTAL EXPENSES",TAB(M2),%9F2,A(35)
28400 A(36) = A(18) – A(35)
28500 PRINT "*** NET BUSINESS INCOME",TAB(M3),%9F2,A(36)
28600 A(5) = A(1) + A(2) + A(3) + A(4) + A(36)
```

```
28700 PRINT "*** TOTAL INCOME",TAB(M3),%9F2,A(5)
28800 GOTO 29000
28900 IF J1< >50 THEN 30200
29000 PRINT "050 ADJUSTMENTS TO INCOME",TAB(M3),%9F2,A(6)
29100 IF M1=2 THEN 29400
29200 INPUT A(6)
29300 IF M1=1 THEN 18800
29400 A(7)=A(5)-A(6)
29500 PRINT "***ADJUSTED GROSS INCOME",TAB(M3),%9F2,A(7)
29600 PRINT
29700 IF M1< >0 THEN 30300
29800 INPUT "WANT TO ITEMIZE DEDUCTIONS? ",A4$
29900 IF A4$="N" THEN 36700
30000 IF A4$< >"Y" THEN 29800
30100 GOTO 30300
30200 IF J1< >200 THEN 30800
30300 PRINT "200 MEDICAL INSURANCE",TAB(M2),%9F2,A(40)
30400 IF M1=2 THEN 30900
30500 INPUT A(40)
30600 IF M1=1 THEN 18800
30700 GOTO 30900
30800 IF J1< >210 THEN 31400
30900 PRINT "210 MEDICINE AND DRUGS",TAB(M2),%9F2,A(41)
31000 IF M1=2 THEN 31500
31100 INPUT A(41)
31200 IF M1=1 THEN 18800
31300 GOTO 31500
31400 IF J1< >220 THEN 33300
31500 PRINT "220 OTHER MEDICAL EXPENSE",TAB(M2),%9F2,A(42)
31600 IF M1=2 THEN 31900
31700 INPUT A(42)
31800 IF M1=1 THEN 18800
31900 C1=A(41)/2
32000 IF C1>150 THEN C1=150
32100 C2=A(7)*.01
32200 C3=A(41)-C2
32300 IF C3<0 THEN C3=0
32400 C4=A(40)-C1
32500 C5=C3+A(40)-C1+A(42)
32600 C6=A(7)*.03
32700 C7=C5-C6
32800 IF C7<0 THEN C7=0
32900 A(43)=C1+C7
33000 PRINT "*** MEDICAL DEDUCTION",TAB(M2),%9F2,A(43)
33100 IF M1=2 THEN 33400
33200 GOTO 33400
33300 IF J1< >240 THEN 33900
33400 PRINT "240 STATE AND LOCAL TAX",TAB(M2),%9F2,A(44)
33500 IF M1=2 THEN 34000
33600 INPUT A(44)
33700 IF M1=1 THEN 18800
33800 GOTO 34000
33900 IF J1< >250 THEN 34500
34000 PRINT "250 OTHER TAXES",YAB(M2),%9F2,A(45)
34100 IF M1=2 THEN 34600
34200 INPUT A(45)
34300 IF M1=1 THEN 18800
34400 GOTO 34600
34500 IF J1< >260 THEN 35100
34600 PRINT "260 INTEREST EXPENSE",TAB(M2),%9F2,A(46)
34700 IF M1=2 THEN 35200
34800 INPUT A(46)
34900 IF M1=1 THEN 18800
35000 GOTO 35200
35100 IF J1< >270 THEN 35700
35200 PRINT "270 CONTRIBUTIONS",TAB(M2),%9F2,A(47)
35300 IF M1=2 THEN 35800
35400 INPUT A(47)
35500 IF M1=1 THEN 18800
35600 GOTO 35800
35700 IF J1< >280 THEN 36300
35800 PRINT "280 CASUALTY OR THEFT",TAB(M2),%9F2,A(48)
35900 IF M1=2 THEN 36400
36000 INPUT A(48)
36100 IF M1=1 THEN 18800
36200 GOTO 36400
36300 IF J1< >290 THEN 36800
36400 PRINT "290 MISC DEDUCTIONS",TAB(M2),%9F2,A(49)
36500 IF M1=2 THEN 36800
36600 INPUT A(49)
36700 IF M1=1 THEN 18800
36800 A(50)=A(43)+A(44)+A(45)+A(46)+A(47)+A(48)+A(49)
36900 PRINT "*** ITEMIZED DEDUCTIONS",TAB(M3),%9F2,A(50)
37000 D=A(7)*.16
37100 IF D>1700 THEN A(9)=D ELSE A(9)=1700
37200 IF A(9)>2400 THEN A(9)=2400
37300 PRINT "*** STANDARD DEDUCTION",TAB(M3),%9F2,A(9)
37400 IF A(9)>A(50) THEN 37800
37500 PRINT "*** ITEMIZED DEDUCTIONS USED"
37600 A(9)=A(50)
37700 GOTO 37900
37800 PRINT "*** STANDARD DEDUCTIONS USED"
37900 A(10)=A(8)*750
38000 PRINT "*** STANDARD EXEMPTION",TAB(M3),%9F2,A(10)
38100 A(11)=A(7)-A(9)-A(10)
38200 PRINT "*** STANDARD EXEMPTION",TAB(M3),%9F2,A(11)
38300 PRINT
38400 REM TAX CREDITS
38500 C1=35*A(8)
38600 C2=.02*A(11)
38700 IF C2>180 THEN C2=180
38800 IF C2<C1 THEN C2=C1
38900 PRINT "*** TAX CREDIT",TAB(M3),%9F2,C2
39000 GOTO 39200
39100 IF J1< >86 THEN 40700
39200 PRINT "086 OTHER TAXES OWED",TAB(M3),%9F2,A(12)
39300 IF M1=2 THEN 39600
39400 INPUT A(12)
39500 IF M1=1 THEN 18800
39600 G1=A(11)
39700 GOSUB 48600
39800 A(29)=T1
39900 PRINT "*** FEDERAL TAX",TAB(M3),%9F2,A(29)
40000 GOSUB 50300
40100 A(30)=T3
40200 PRINT "***F.I.C.A.",TAB(M3),%9F2,A(30)
40300 A(13)=A(29)+A(12)+A(30)-C2
40400 PRINT "*** TOTAL FEDERAL TAXES OWED",TAB(M3),%9F2,A(13)
40500 PRINT
40600 GOTO 40800
40700 IF J1< >90 THEN 41300
40800 PRINT "090 FEDERAL TAX WITHHELD",TAB(M3),%9F2,A(14)
40900 IF M1=2 THEN 41400
41000 INPUT A(14)
41100 IF M1=1 THEN 18800
41200 GOTO 41400
41300 IF J1< >95 THEN 41900
41400 PRINT "095 ESTIMATED TAX PAYMENTS",TAB(M3),%9F2,A(15)
41500 IF M1=2 THEN 42000
41600 INPUT A(15)
41700 IF M1=1 THEN 18800
41800 GOTO 42000
41900 IF J1< >97 THEN 43600
42000 PRINT "097 F.I.C.A. WITHHELD",TAB(M3),%9F2,A(27)
42100 IF M1=2 THEN 42400
42200 INPUT A(27)
42300 IF M1=1 THEN 18800
42400 A(16)=A(15)+A(14)+A(27)
42500 PRINT "*** TOTAL TAX PAYMENTS",TAB(M3),%9F2,A(16)
42600 PRINT
42700 C1=A(16)-A(13)
42800 IF C1<0 THEN 43200
42900 PRINT "*** IRS OWES YOU ",TAB(M3),%9F2,C1
43000 PRINT
43100 GOTO 43700
43200 C1=ABS(C1)
43300 PRINT "*** YOU OWE IRS ",TAB(M3),%9F2,C1
43400 PRINT
43500 GOTO 43700
43600 IF J1< >98 THEN 44200
43700 PRINT "098 CALIF TAX WITHHELD",TAB(M3),%9F2,A(28)
43800 IF M1=2 THEN 44300
43900 INPUT A(28)
44000 IF M1=1 THEN 18800
44100 GOTO 44300
44200 IF J1< >93 THEN 47000
44300 PRINT "093 CALIF ESTIMATED PAYMENTS",TAB(M3),%9F2,A(32)
44400 IF M1=2 THEN 44800
44500 INPUT A(32)
44600 IF M1=1 THEN 18800
44700 REM CALIFORNIA STANDARD DEDUCTIONS
44800 C1=1000*A(8)
44900 G1=A(5)-C1
45000 PRINT "*** CALIF STANDARD DEDUCTIONS",TAB(M3),%9F2,C1
45100 PRINT "*** CALIF TAXABLE INCOME",TAB(M3),%9F2,G1
45200 GOSUB 49500
45300 A(31)=T2
45400 A(33)=A(32)+A(28)
45500 PRINT "*** TOTAL CALIF TAX PAID",TAB(M3),%9F2,A(33)
45600 PRINT "*** TOTAL CALIF TAX OWED",TAB(M3),%9F2,A(31)
45700 C1=A(33)-A(31)
45800 IF C1>0 THEN 46100
45900 PRINT "*** OWE CALIF TAX",TAB(M3),%9F2,ABS(C1)
46000 GOTO 46200
46100 PRINT "*** REFUND CALIF TAX",TAB(M3),%9F2,C1
46200 PRINT   PRINT   PRINT
46300 PRINT "*** FED TAX % OF INCOME",TAB(M3,%9F2,A(13)/A(7)*100
46400 PRINT "*** CALIF TAX % OF INCOME",TAB(M3),%9F2,A(31)/A(7)*100
46500 PRINT "*** F.I.C.A. % OF INCOME",TAB(M3),%9F2,A(30)/A(7)*100
46600 T3=A(13)+A(31)+A(30)
46700 PRINT "*** TOTAL TAX % OF INCOME",TAB(M3),%9F2,T3*100/A(7)
46800 PRINT   PRINT
46900 IF M1< >1 THEN 47200
47000 PRINT "LINE NO.",%4I,J1," NOT RECOGNIZED"
47100 GOTO 18800
47200 IF M1=2 THEN 51300
47300 INPUT "DO YOU WANT TO SAVE THE RESULTS? ",A4$
47400 IF A4$="N" THEN 48200
47500 IF A4$< >"Y" THEN 47300
47600 INPUT "OUTPUT FILE NAME? ",A4$
47700 OPEN #0,A4$
47800 FOR J=1 TO K1
47900 WRITE #0,A(J)
48000 NEXT J
48100 CLOSE #0
48200 INPUT "DO YOU WANT A LISTING? ",A4$
48300 IF A4$="N" THEN 51300
48400 IF A4$< >"Y" THEN 48200
48500 GOTO 17900
48600 REM ACTUAL CALCULATION SUBROUTINE
48700 REM FEDERAL TAX CALCULATION
48800 FOR J=1 TO 18
48900 IF G1>T1(2,J) THEN 49200
49000 T1=T1(3,J)+(G1-T1(1,J))*T1(4,J)*.01
49100 GOTO 49400
49200 NEXT J
49300 PRINT "AMOUNT NOT IN FED TAX TABLE",G1
49400 RETURN
49500 REM STATE TAX CALCULATIONS
49600 FOR J=1 TO 11
49700 IF G1>T2(2,J) THEN 50000
49800 T2=T2(3,J)+G1-T2(1,J))*T2(4,J)*.01
49900 GOTO 50200
50000 NEXT J
50100 PRINT "AMOUNT NOT IN CALIF TAX TABLE",G1
50200 RETURN
50300 REM FICA CALCULATION
50400 G1=A(1)
50500 IF G1>16500 THEN G1=16500
50600 T3=G1*.059
50700 IF G1=16500 THEN RETURN
50800 G2=G1+A(36)
50900 IF G2>16500 THEN G2=16500
51000 G2=G2-G1
51100 T3=T3+G2*.079
51200 RETURN
51300 INPUT "WANT ANOTHER RUN? ",A4$
51400 IF A4$="Y" THEN 51600
51500 STOP
51600 A1$="OLD"
51700 GOTO 16700
READY
```

Hamming's Error Corrections

By Ed Comer

Imagine that you have loaded your micro system, using the new Kansas City Standard, when the program bombs because of tape read errors. Frustrating? This is not a new problem, nor one restricted to computer hobbyists. This problem has been experienced by most people inputting data from a magnetic medium, and the cassettes that hobbyists use will amplify the problem. Someone else can expound upon *how* the data recording method or equipment created the errors, I would like to talk about getting rid of them. In fact, instead of rehashing how to detect errors (CRC, CHECKSUM, etc.), I want to explore a method of *corrective action.* In other words, let's correct errors as they are read in, in order to avoid rewinding the tape if an error is detected.

After errors are detected, in most systems, the normal action is to rewind the tape and start over. If no additional errors occur, the load will complete. repeated errors could turn a simple program into an all-day affair. In situations where a slow input medium such as the KC Standard is being used, simply detecting errors saves little time over a complete load, unless your luck is with you.

The first problem is to detect the errors. The simplest method is to load and run. If the load runs, so far so good. If it does not run then you have detected an error. It would be desirable to avoid a complete load before errors are detected. Hence enter the error checking programs previously mentioned. These programs would record the data in a prescribed format, and upon playback check for the same format as was recorded. The methods most commonly used are (1) a per-byte parity check, (2) a word block math checksum, (3) a word block logically generated CRC cycle redundancy check. As I said, these methods are much used and detailed explanations of them can be found in many references. For the purposes of this article let's pick simple PARITY as our error-detecting method.

I do not advocate anything new to the computer world, only that I suggest the micro-software adaptation of big machine hardware that has existed for many years.

The parity bit will detect an error on the total word, and allow the program to flag that word as containing an error. Now the problem is: which bit is in error? This problem was solved by a Bell Labs mathematician named Hamming, in the 1960's. Hamming's idea was to add parity bits to the word so that each parity bit checks only a part of the word. Whenever an error is detected, the mathematical combination of these "Hamming bits" will allow the correction of the bit in error.

We will take our 8-bit word and add three Hamming bits and one parity bit to it. This will require splitting up the word as shown in Figure 1. The data bits are a to e, while each 8-bit word has two sets of Hamming and parity bits, indicated by X and P. The fact that each 8-bit word will take two words to express it, doubling the load time, is offset by the fact that each load will be error free.

To write the data word onto the tape the program will code the data bytes as shown in Figure 2. To illustrate, let's code the 8080 CALL instruction (315) as it would be done by this Hamming program.

The X1 Hamming bit is made a zero or one to give odd parity to bits 0, 2 and 3. This can be seen by examining Figure 2. Thus for our example "315", x1 = 0. This is because the first word contains bits ABCD which are 1101. A = 1,B is not looks at by X1, C = 1, and D = 1. This is an odd number of ones, so X1 is made zero to keep the number of ones odd.

The X2 Hamming bit checks bits A,B and d. A = 1, B = 0, C is not looked at by X2, D = 1. This is an even number of ones, so X2 is made a 1 to give an odd number of ones for this check.

The X4 Hamming bit checks bits A, B and C. A = 1, B = 0, C = 1, so X4 = 1. Now the first portion of the job is completed by setting the parity bit to equal odd parity for the word. Out first word of the format used in Figure 1 is 00111101. The remaining high-order bits of our example are encoded in the same manner into the second word. The word pair is recorded onto the magnetic tape, and on replay can be reassembled into "315".

7	6	5	4	3	2	1	0	BIT POS.
P	X_1	X_2	D	X_4	C	B	A	1ST WORD
P	X_1	X_2	H	X_4	G	F	E	2ND WORD
PARITY	HAMMING				BYTE			

Figure 1.

7	6	5	4	3	2	1	0	BIT POSITION
	1	2	3	4	5	6	7	CORRECTION CODE POSITION
P	X_1	X_2	D	X_4	C	B	A	CODE ELEMENT
	X		X		X		X	X_1 CHECK ELEMENTS
		X	X			X	X	X_2 CHECK ELEMENTS
				X	X	X	X	X_4 CHECK ELEMENTS
	X	X	X	X	X	X	X	P

Figure 2.

word 1 00111101
word 2 01010100 = 315

Now that all the data are recorded in our Hamming format, we can reverse the operation to read it. Our tape *read* program will use the logic shown in Figure 3 to reassemble the data into their original form. All the Hamming information will only be used for error correction, and is discarded and not loaded into memory.

In the flowchart there is one process block that needs further explanation that is titled "CORRECT BAD BIT." Refer to Figure 4 for a flowchart of the "CORRECT BAD BIT" routine.

Assume that bit 2 of word 1, of our example, is in error. By going through Figure 4, you will see that the correction code value will equal 5. The program will then "FLIP" the bit associated with correction code value 5 according to Figure 2. Thus whatever the incorrect value was, it is now corrected, and the tape goes on!

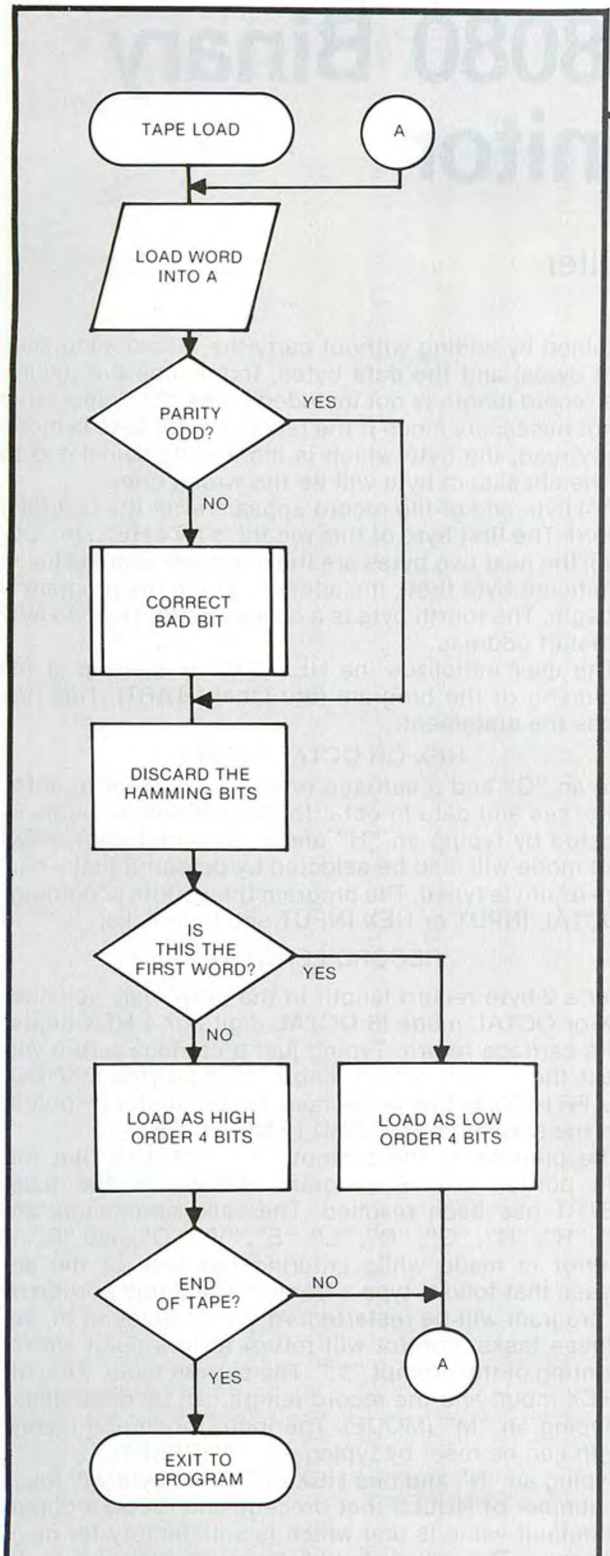


Figure 3.

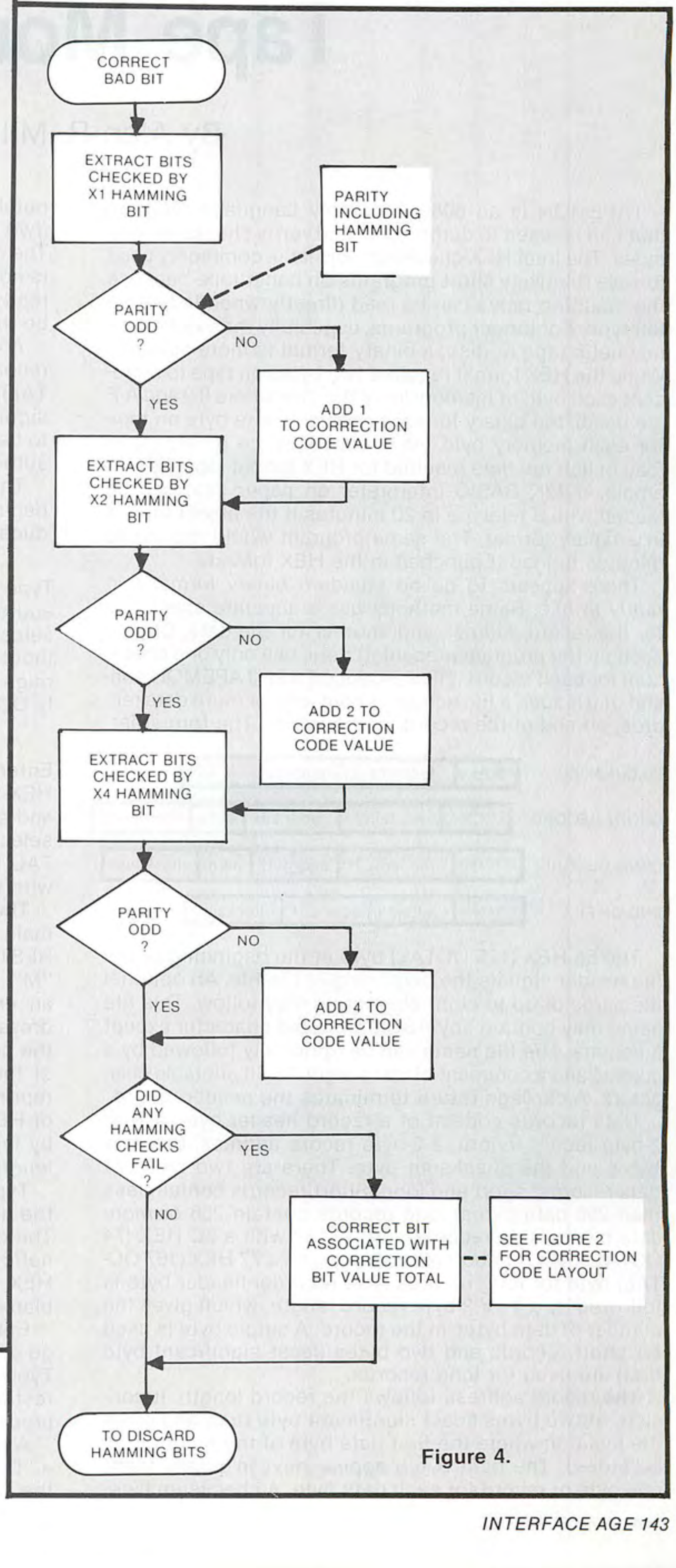


Figure 4.

TAPEMON: An 8080 Binary Tape Monitor

By Alan R. Miller

TAPEMON is an 8080 Assembly Language program that can be used to dump, load, and verify checksummed tapes. The Intel HEX checksum format is commonly used to save relatively short programs on paper tape because the resulting tapes can be read directly when fed into a teletype. For longer programs, especially those saved on magnetic tape or disc, a binary format is more suitable. While the HEX format requires two bytes on tape to represent each byte of memory (only the characters 0-9 and A-F are used), the binary format needs only one byte on tape for each memory byte. As a consequence binary tapes load in half the time required for HEX format tape. For example, a 12K BASIC interpreter on paper tape can be loaded with a teletype in 20 minutes if the object tape is in a binary format. The same program would require 40 minutes to load if punched in the HEX format.

There appears to be no standard binary format currently in use. Some methods use a separate checksum for the record address and another for the data. Others, such as the program presented here, use only one checksum for each record. Files produced with TAPEMON consist of a leader, a file header record, one or more data records, an end-of-file record, and a trailer. The format is:

FILEHEADER	55H	filename , comment CR			
SHORT RECORD	3CH	rec len	addr L,H	data	checksum
LONG RECORD	77H	rec len L,H	addr L,H	data	checksum
END-OF-FILE	74H	autostart addr L,H	checksum		

The 55 HEX (125 OCTAL) byte at the beginning of the file header signals the beginning of the file. An optional file name of up to eight characters may follow. This file name may contain any ASCII printable character except a comma. The file name can be optionally followed by a comma and a comment of up to eight ASCII printable characters. A carriage return terminates the header record.

Data records consist of a record header byte, a 1- or 2-byte record length, a 2-byte record address, the data bytes and the checksum byte. There are two types of data records: short and long. Short records contain less than 256 data bytes; long records contain 256 or more data bytes. Data records start either with a 3C HEX (74 OCTAL) byte for short records, or with a 77 HEX (167 OCTAL) byte for long records. The recorder-header byte is followed by a 1- or 2-byte record length, which gives the number of data bytes in the record. A single byte is used for short records and two bytes (least significant byte first) are used for long records.

The record address follows the record length. It consists of two bytes (least significant byte first) and gives the location where the first data byte of the record is to be stored. The data bytes appear next in binary form, one byte of record for each data byte. A checksum byte, obtained by adding without carry the record addresses (two bytes) and the data bytes, terminates the record. The record length is not included in the checksum. This is not necessary since if the record-length byte is incorrectly read, the byte which is incorrectly calculated to be the checksum byte will be the wrong one.

A 4-byte end-of-file record appears after the last data record. The first byte of this record is a 74 HEX (164 OCTAL), the next two bytes are the autostart address (least significant byte first), the address where the program is to begin. The fourth byte is a checksum for the two-byte autostart address.

The user initializes the HEXMON by starting at the beginning of the program (the label START). This produces the statement:

HEX OR OCTAL INPUT?

Type an "O" and a carriage return if you want to enter addresses and data in octal format. HEX-input mode is selected by typing an "H" and a carriage return. HEX-input mode will also be selected by default if just a carriage return is typed. The program then prints accordingly OCTAL INPUT or HEX INPUT and then asks:

RECORD LENGTH?

Enter a 2-byte record length in the previously selected HEX or OCTAL mode (6 OCTAL digits or 4 HEX digits) and a carriage return. Typing just a carriage return will select the default record length of 255 bytes (377 OCTAL, FF HEX). In this latter case, the computer responds with the statement RECORD LENGTH 255.

The printing of the prompt ">:" indicates that the main portion of the program starting at the label RESTRT has been reached. The valid commands are "M", "R", "N", "G", "D", "L", "E", "V", "O", and "C". If an error is made while entering the task or the addresses that follow, type a Control-X and this portion of the program will be restarted. At the completion of any of these tasks, control will return to this point with a reprinting of the prompt ">:". The current mode (OCTAL or HEX input) and the record length can be determined by typing an "M" (MODE). The input mode and record length can be reset by typing an "R" (RESET).

Typing an "N" and one HEX or OCTAL byte will reset the number of NULLS that precede and follow a dump. The default value is one which is satisfactory for magnetic tape. The value should, however, be reset to 48 HEX (110 OCTAL) for paper tape. This will produce a blank 6-inch leader and trailer.

Enter a "G", a 2-byte address, and a carriage return to go somewhere else, for example to your regular monitor. Typing a Control-X during input will cancel the line and restart this program. A "WHAT?" will be printed for improper input (e.g., HEX characters when in OCTAL mode).

A portion of memory can be dumped to tape by typing a "D" (for DUMP), the start address, the stop address, the autostart address (two bytes for each address), and

a carriage return. An optional file name of up to eight characters can be typed after the autostart address and before the carriage return. Any printable ASCII character except a comma can be used. If a file name is entered, then an optional comment of up to eight characters can also be used. The comment can contain any printable ASCII character, and is separated from the file name by a comma. For example:

>:F800:F91F:F803:PROM4,VER 4.1 <CR>

will dump from F800 to F91F with an autostart address of F803. The header will carry the file name/comment PROM,VER 4.1. (The colons and > symbol are printed by the program).

If an error is made during entry of the file name or comment, type a DEL (RUB OUT) and then the correct character. A back arrow is echoed when the DEL key is pressed.

A tape can be loaded by typing an "L" and a carriage return. The keyboard bell will ring and the front panel lights (if you have them) will change when the file header (55 HEX) is found. This feature requires six additional bytes and an additional tape-input routine, one is used to look for the file header and the other is used for everything else. I feel that the additional complexity is worth it since I don't have to wait until a tape has been played through to find out that I have the left channel plugged into the computer, but the program I want is on the right channel.

When the tape has successfully loaded, the autostart address will be printed in both HEX and OCTAL. The prompt ">:" will be printed indicating that the program is ready for the next task. Another way to load a tape is to type the file name after the "L" command, e.g.

>:LPROM4 <CR>

can be used to load the tape made in the tape made in the above example. The DEL key can be used to correct errors made while entering the file name. If the file name in the command line does not match the file name on the tape, the task is terminated. The program prints an error statement followed by the actual file name and comment, e.g.

WRONG FILE NAME, TRY: PROM4,VER 4.1

would be printed if the incorrect file name LPROM3 were given. If the program loads correctly, the file name, comments and autostart address are printed.
For example:

PROM4,VER 4.1 STARTS AT F803:370003

would appear in this sample.

A program can be loaded and executed by typing an "E", optionally a file name and a carriage return. When the program has been loaded, the program counter will jump to the autostart address.

A tape can be loaded at other than its normal address by typing an "O" (for offset), a 2-byte offset address (in the current OCTAL or HEX mode), optionally the file name and a carriage return. An offset of 0400 HEX (004000 OCTAL) will load the program 1K bytes (1024 bytes) higher than the normal address. The program can be loaded 4K bytes lower than the normal address with an offset of either F000 HEX (360000 OCTAL) or -1000 HEX (-020000 OCTAL). If for example an offset of F000 is added to the program address of 3000, the double register add gives an address of 2000. A negative offset value is first subtracted from zero and then added to the program address. During the offset load, the original address is added to the checksum so that it is properly calculated. Of course the jumps and calls are not altered, so that the program will not run at the new location if there are jumps and calls addressed for original location.

MITS software such as BASIC and the Software Package II assembler is provided on paper or magnetic tape in a binary format that is compatible with TAPEMON. This software, however, also contains a reverse-loaded checksum loader ahead of the main program. All MITS programs can be loaded with the command "C" (for checksum) and a carriage return. The checksum loader at the beginning of the tape is scanned for the disable interrupt command (DI). This DI command is the last byte of the checksum loader and now represents the file header. The record header byte of 3C HEX following the checksum loader is then searched for. On the other hand, absolute tapes made with the MITS Software Package II monitor itself are fully compatible with TAPEMON and can be loaded with the "L", "E", or "O" commands. MITS does not use a checksum on the end-of-file record, but since the record header is a 78 HEX (170 OCTAL) rather than a 74 HEX, TAPEMON can distinguish between the two types of EOF records.

Tapes loaded with TAPEMON do not have to be verified since they are checksummed. If the load operation was completed, the tape was loaded correctly. Tapes dumped with TAPEMON of course should be verified to be certain that they were properly recorded. A defective spot on the tape for example may give an error. Tapes can be verified by playing back the tape, typing "V", optionally the file name, and a carriage return.

For all of the above load and verify operations, the two record-address bytes and the data bytes are summed and compared to the checksum value at the end of each record. If the two do not match, the operation is terminated, and the message:

CHECKSUM ERROR AT F810:370008

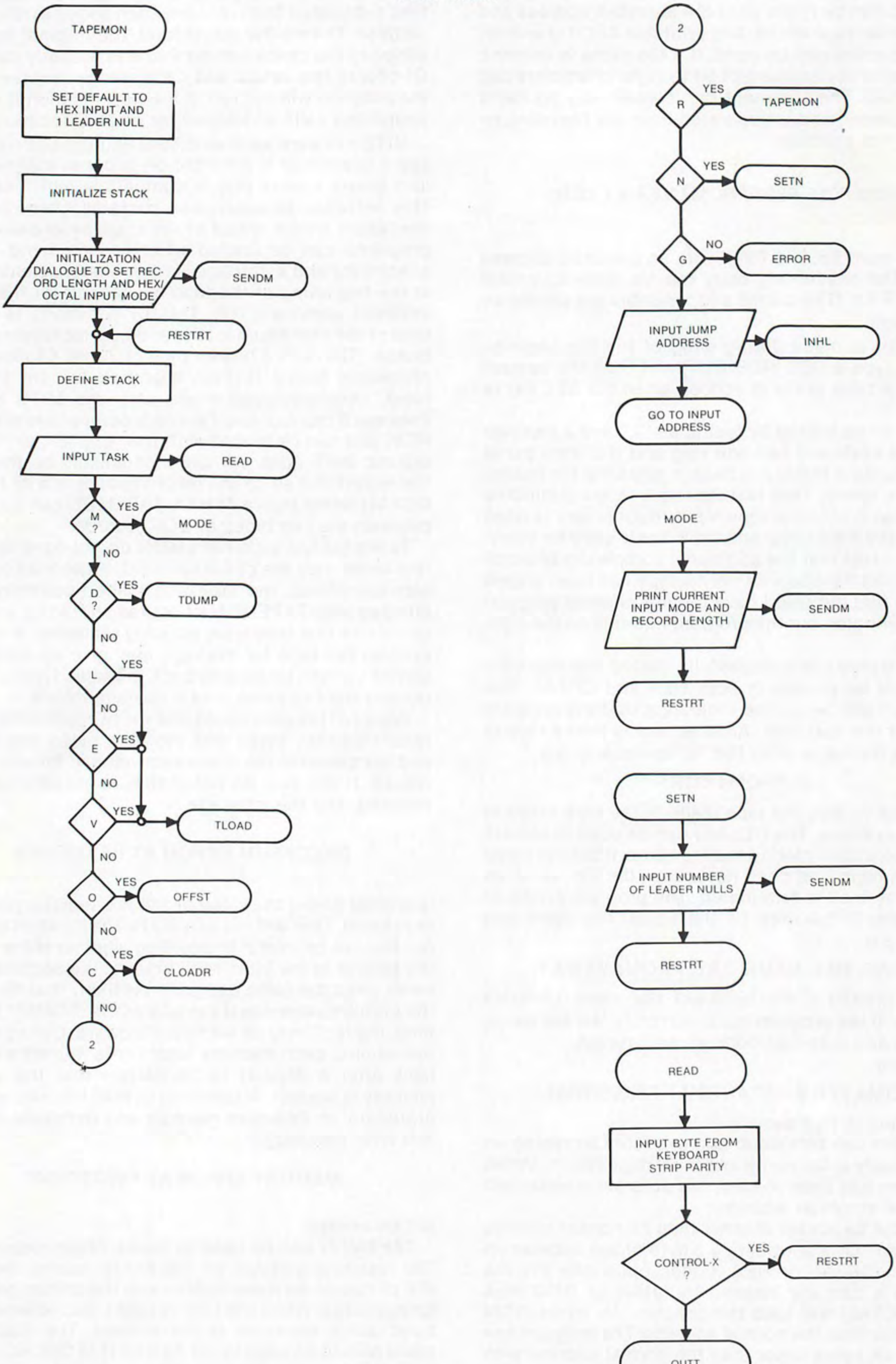
is printed giving the value of the H,L register pair at the checksum. This will not usually be the location of the error, but can be useful in deciding whether the error is in the tape or in the interface circuitry. If a second load or verify gives the same address, it is likely that the tape is the problem, whereas if the address is different the next time, the fault may lie with the hardware. During the load operations, each memory location is immediately read back after a deposit to be certain that the value in memory is correct. Attempting to load into non-existent, protected, or defective memory will terminate the load and error message:

MEMORY ERROR AT F820:370040

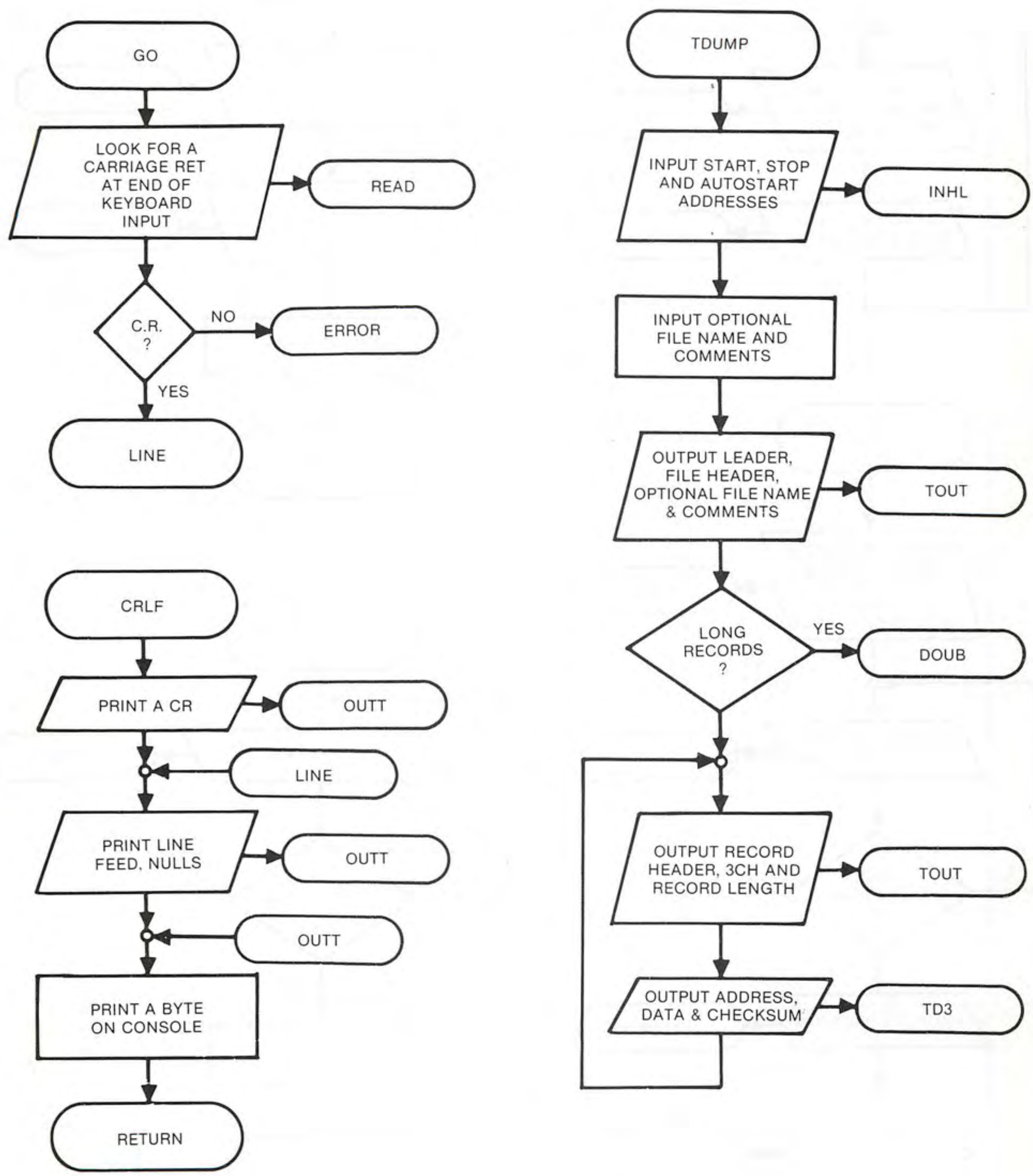
will be printed.

TAPEMON can be used to punch binary paper tapes. The resulting garbage on the printer during the dump will of course be meaningless and the printer will make funny noises. When the tape is read back, however, all is quiet, since the input is not echoed. The NULL command should be used to set 48 HEX (110 OCTAL) nulls to provide a leader and trailer of six inches. If paper tape is the usual medium, change the default option to provide 64 nulls during initialization.

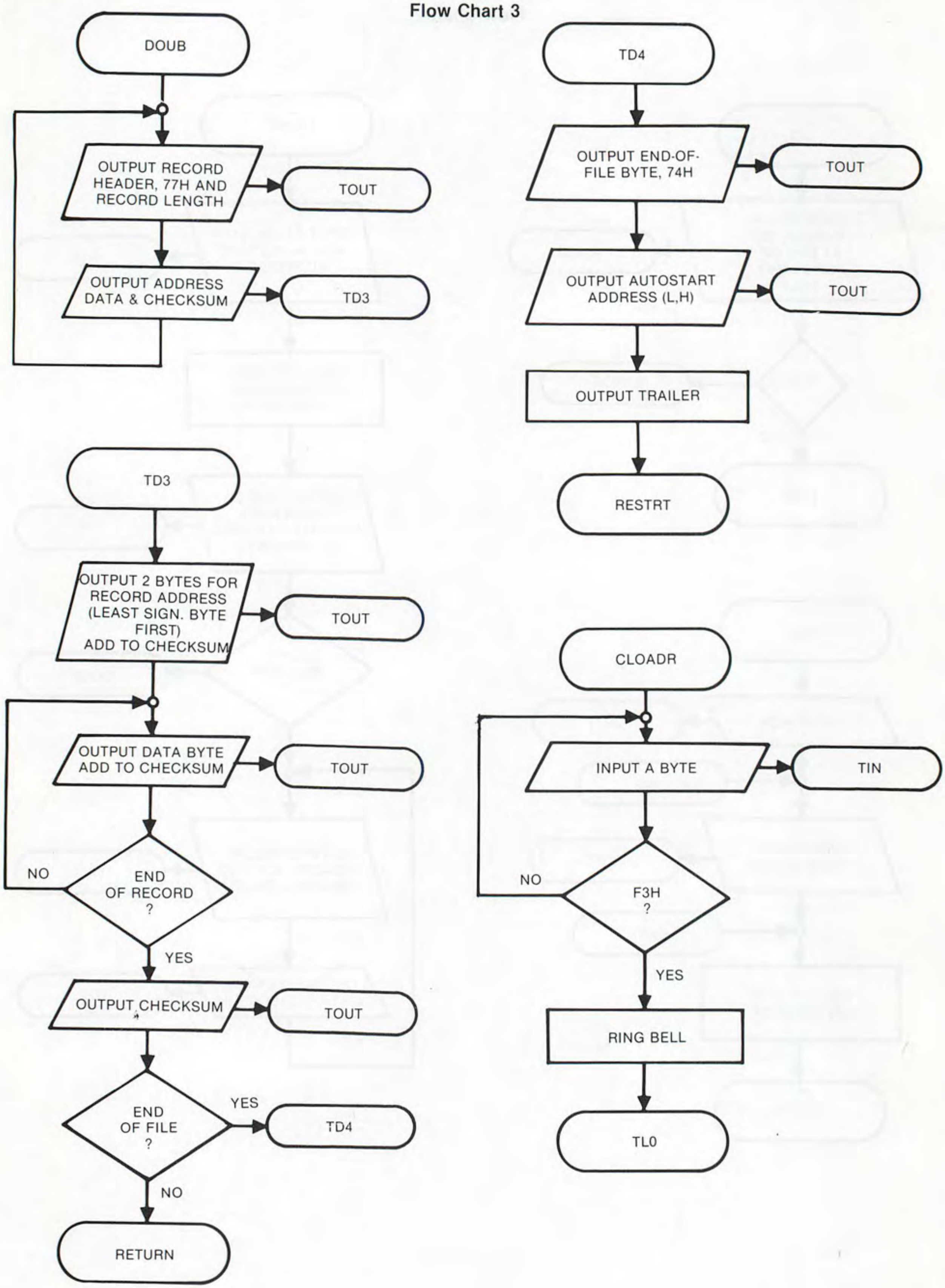
Flow Chart 1

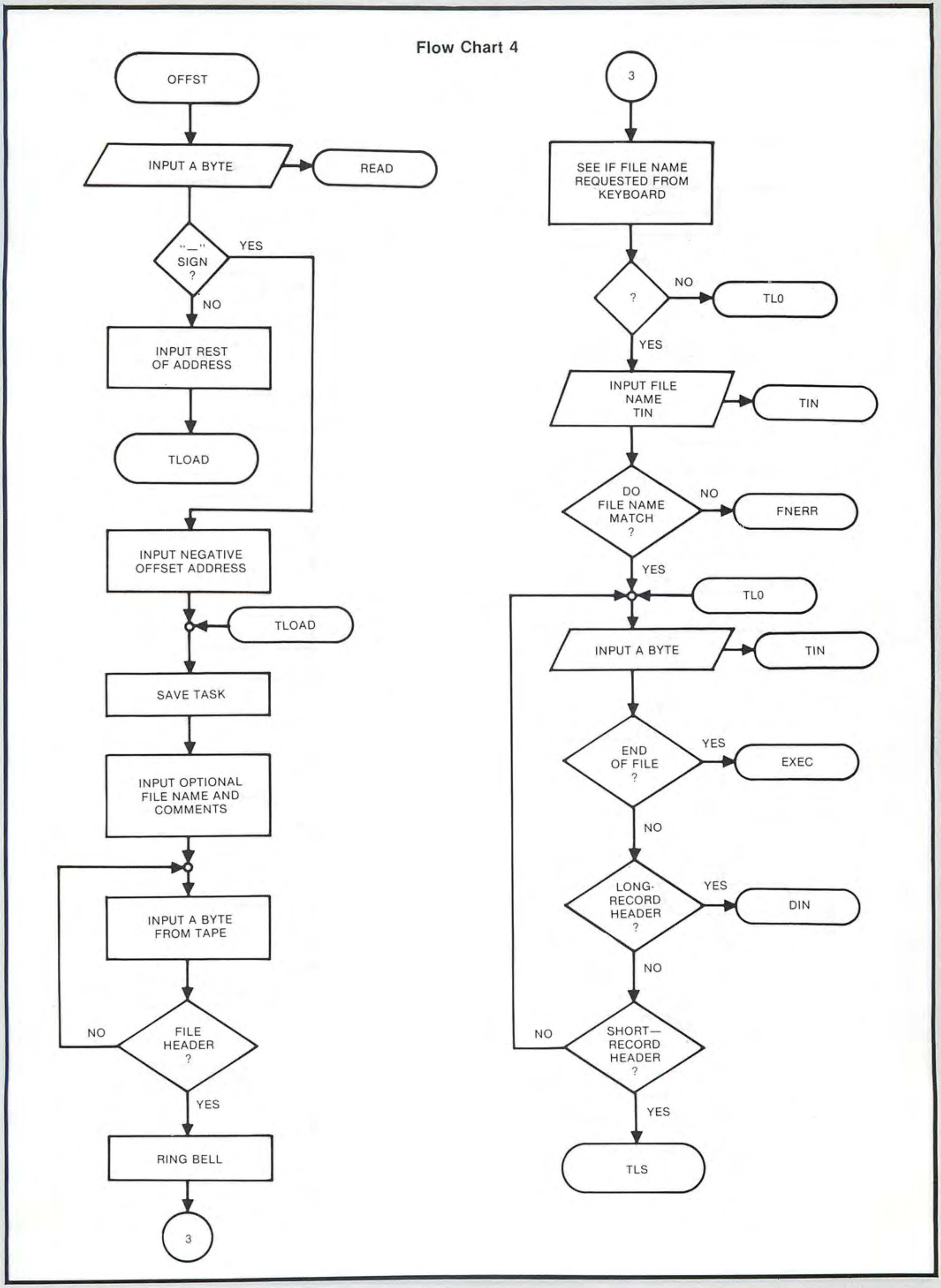

TAPEMON
SET DEFAULT TO HEX INPUT AND 1 LEADER NULL
INITIALIZE STACK
INITIALIZATION DIALOGUE TO SET RECORD LENGTH AND HEX/OCTAL INPUT MODE
SENDM
RESTRT
DEFINE STACK
INPUT TASK
READ
M ?
YES
MODE
NO
D ?
TDUMP
L
E
V
TLOAD
O
OFFST
C
CLOADR
2
R
TAPEMON
N
SETN
G
ERROR
INPUT JUMP ADDRESS
INHL
GO TO INPUT ADDRESS
MODE
PRINT CURRENT INPUT MODE AND RECORD LENGTH
SENDM
RESTRT
SETN
INPUT NUMBER OF LEADER NULLS
SENDM
RESTRT
READ
INPUT BYTE FROM KEYBOARD STRIP PARITY
CONTROL-X ?
RESTRT
OUTT

Flow Chart 2

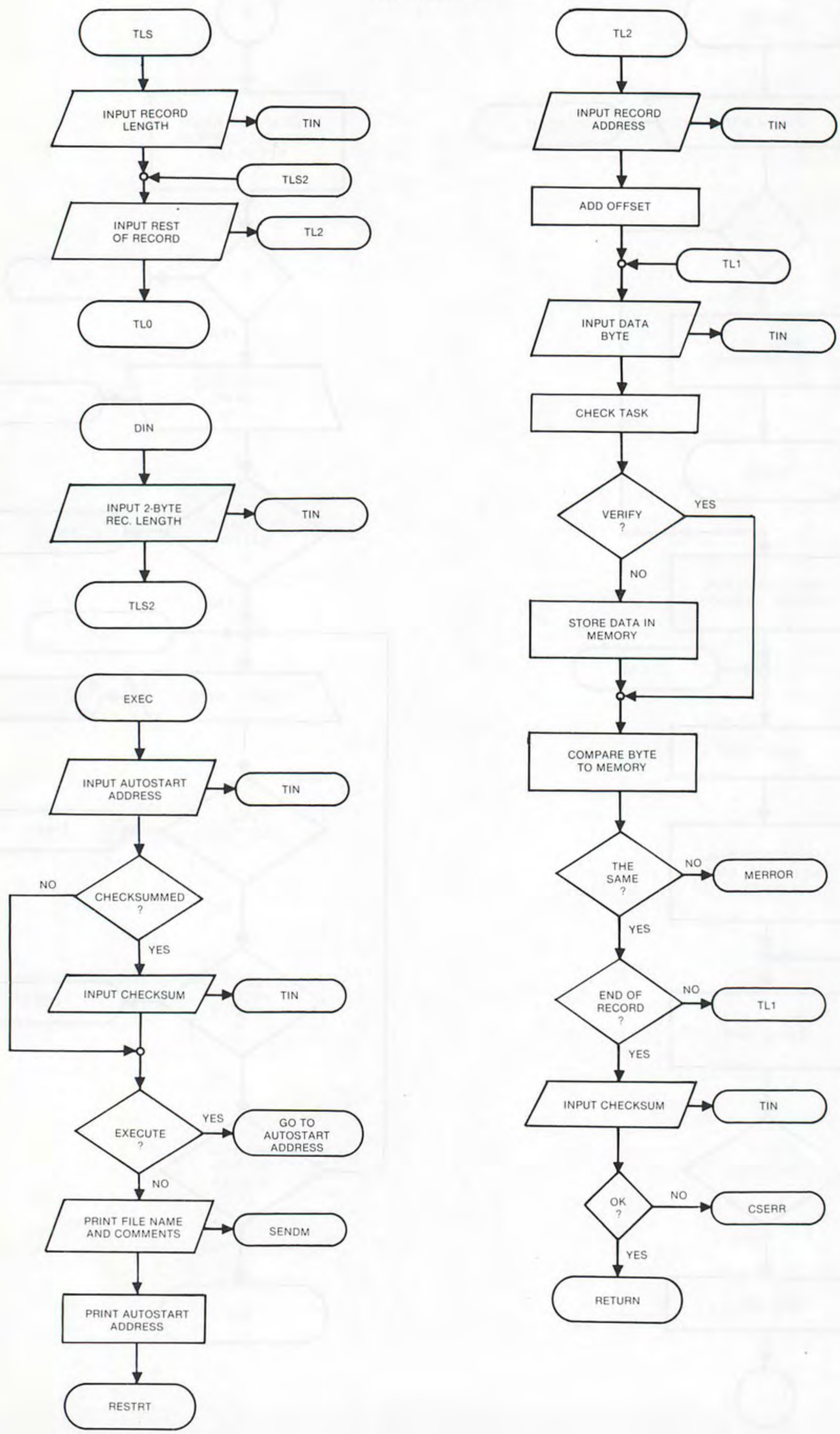

GO
LOOK FOR A CARRIAGE RET AT END OF KEYBOARD INPUT
READ
C.R. ?
NO
ERROR
YES
LINE
CRLF
PRINT A CR
OUTT
LINE
PRINT LINE FEED, NULLS
OUTT
OUTT
PRINT A BYTE ON CONSOLE
RETURN
TDUMP
INPUT START, STOP AND AUTOSTART ADDRESSES
INHL
INPUT OPTIONAL FILE NAME AND COMMENTS
OUTPUT LEADER, FILE HEADER, OPTIONAL FILE NAME & COMMENTS
TOUT
LONG RECORDS ?
YES
DOUB
OUTPUT RECORD HEADER, 3CH AND RECORD LENGTH
TOUT
OUTPUT ADDRESS, DATA & CHECKSUM
TD3

Flow Chart 3

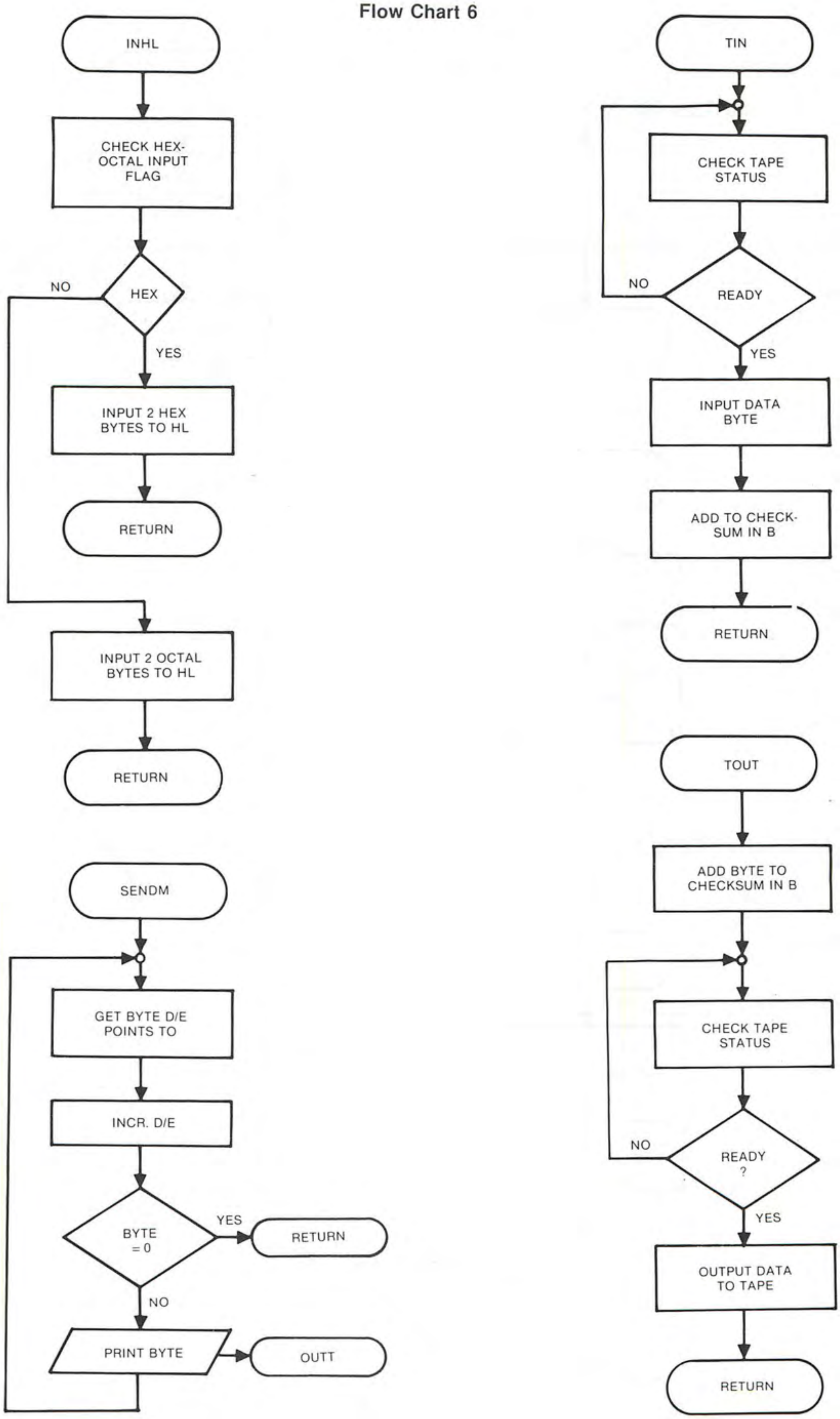

DOUB
OUTPUT RECORD HEADER, 77H AND RECORD LENGTH
TOUT
OUTPUT ADDRESS DATA & CHECKSUM
TD3
TD3
OUTPUT 2 BYTES FOR RECORD ADDRESS (LEAST SIGN. BYTE FIRST) ADD TO CHECKSUM
TOUT
OUTPUT DATA BYTE ADD TO CHECKSUM
TOUT
NO
END OF RECORD ?
YES
OUTPUT CHECKSUM
TOUT
END OF FILE ?
YES
TD4
NO
RETURN
TD4
OUTPUT END-OF-FILE BYTE, 74H
TOUT
OUTPUT AUTOSTART ADDRESS (L,H)
TOUT
OUTPUT TRAILER
RESTRT
CLOADR
INPUT A BYTE
TIN
NO
F3H ?
YES
RING BELL
TLO


Flow Chart 4
OFFST
INPUT A BYTE
READ
"—" SIGN ?
YES
NO
INPUT REST OF ADDRESS
TLOAD
INPUT NEGATIVE OFFSET ADDRESS
TLOAD
SAVE TASK
INPUT OPTIONAL FILE NAME AND COMMENTS
INPUT A BYTE FROM TAPE
FILE HEADER ?
NO
YES
RING BELL
3
3
SEE IF FILE NAME REQUESTED FROM KEYBOARD
?
NO
TL0
YES
INPUT FILE NAME TIN
TIN
DO FILE NAME MATCH ?
NO
FNERR
YES
TL0
INPUT A BYTE
TIN
END OF FILE ?
YES
EXEC
NO
LONG-RECORD HEADER ?
YES
DIN
NO
SHORT—RECORD HEADER ?
NO
YES
TLS

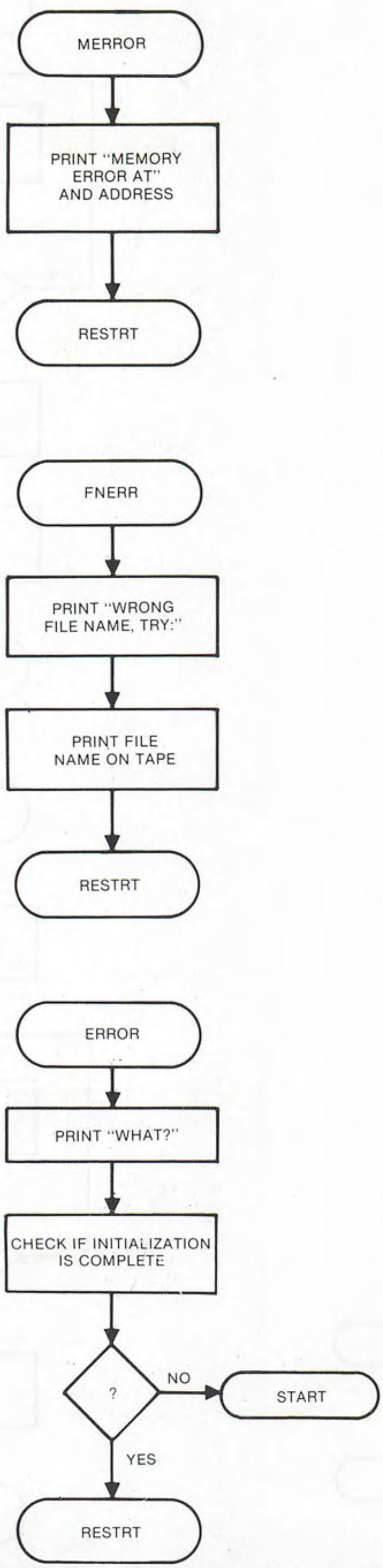
Flow Chart 5


TLS
INPUT RECORD LENGTH
TIN
TLS2
INPUT REST OF RECORD
TL2
TL0
DIN
INPUT 2-BYTE REC. LENGTH
TIN
TLS2
EXEC
INPUT AUTOSTART ADDRESS
TIN
CHECKSUMMED ?
NO
YES
INPUT CHECKSUM
TIN
EXECUTE ?
YES
GO TO AUTOSTART ADDRESS
NO
PRINT FILE NAME AND COMMENTS
SENDM
PRINT AUTOSTART ADDRESS
RESTRT
TL2
INPUT RECORD ADDRESS
TIN
ADD OFFSET
TL1
INPUT DATA BYTE
TIN
CHECK TASK
VERIFY ?
YES
NO
STORE DATA IN MEMORY
COMPARE BYTE TO MEMORY
THE SAME ?
NO
MERROR
YES
END OF RECORD ?
NO
TL1
YES
INPUT CHECKSUM
TIN
OK ?
NO
CSERR
YES
RETURN

Flow Chart 6


INHL
CHECK HEX-OCTAL INPUT FLAG
HEX
NO
YES
INPUT 2 HEX BYTES TO HL
RETURN
INPUT 2 OCTAL BYTES TO HL
RETURN
SENDM
GET BYTE D/E POINTS TO
INCR. D/E
BYTE = 0
YES
RETURN
NO
PRINT BYTE
OUTT
TIN
CHECK TAPE STATUS
NO
READY
YES
INPUT DATA BYTE
ADD TO CHECK-SUM IN B
RETURN
TOUT
ADD BYTE TO CHECKSUM IN B
CHECK TAPE STATUS
NO
READY ?
YES
OUTPUT DATA TO TAPE
RETURN

Flow Chart 7


MERROR
PRINT "MEMORY ERROR AT" AND ADDRESS
RESTRT
CSERR
PRINT "CHECKSUM ERROR AT" AND ADDRESS
RESTRT
FNERR
PRINT "WRONG FILE NAME, TRY:"
PRINT FILE NAME ON TAPE
RESTRT
ERROR
PRINT "WHAT?"
CHECK IF INITIALIZATION IS COMPLETE
?
NO
START
YES
RESTRT

TAPEMON requires 1523 bytes of memory, including twelve levels (24 bytes) of stack. The program is written for the standard MITS configuration of a 2SIO serial port addressed to 10/11 HEX (20/21 OCTAL) and a tape recorder interface addressed to 6/7. The following table gives the locations and parameters that may need to be changed for your system.

	SOURCE PROGRAM VARIABLE	ADDRESS (HEX)	DATA (HEX)
Define stack	STACK	580D,5894	5DEA
Keyboard status	TYSTAT	592D,595D	10
Keyboard data	TYDATA	5934,5965	11
Mask for data avail.	INMASK	592F	01
Mask for output	OUTMSK	595F	02
Jump zero		5930,5960	CA
Tape status	TAPES	5B1F,5C44,5C55	06
Tape data	TAPED	5B26,5C4B,5C5D	07
Mask for data avail.	ACINM	5B21,5C46	01
Mask for tape output	ACOM	5C57	80
Jump not zero		5B22,5C47,5C58	C2
Default record len	RLEN	587A	00FF
Default leader nulls	SNUL	5808	01

PROGRAM LISTING

```
;TAPEMØN:  PRØGRAM TØ LØAD, DUMP AND VERIFY BINARY
;          CHECKSUMMED TAPES WITH AUTØSTART AND
;          WITH A CHØICE ØF HEX ØR ØCTAL INPUT
;
;
;PRØGRAMMED FØR AN 8080 MICRØPRØCESSØR
;BY ALAN R. MILLER
;NEW MEXICØ TECH, SØCØRRØ, NM   87801
; 505-835-5619        SEPTEMBER 4, 1977
;
;REQUIRES 1422 BYTES ØF MEMØRY INCLUDING STACK
;KEYBØARD ADDRESS IS 10/11HEX (20/21 ØCTAL)
;TAPE ADDRESS IS 6/7.  TAPE AND KEYBØARD CAN HAVE
;THE SAME ADDRESS SØ THAT A TELETYPE TAPE CAN BE
;PUNCHED.
;
;WHEN STARTED AT "START", PRØGRAM PRINTS:
;
;       "HEX ØR ØCTAL INPUT?"
;
;TYPE AN "H" FØR HEX MØDE ØR AN "Ø" FØR ØCTAL MØDE
;AND A CARRIAGE RETURN.  (A CARRIAGE RETURN WITHØUT
;AN "H" ØR "Ø" DEFAULTS TØ HEX MØDE.)  THE PRØGRAM
;THEN PRINTS:
;
;       "RECØRD LENGTH?"
;
;TYPE A TWØ-BYTE HEX ØR ØCTAL RECØRD LENGTH AND A
;CARRIAGE RETURN.  (A CARRIAGE RETURN WITH NØ ØTHER
;INPUT DEFAULTS TØ A RECØRD LENGTH ØF 255.)   A PRØMPT
;ØF ">:" IS THEN PRINTED.  THE VALID CØMMANDS ARE:
;"M","R","G","D","L","E","V","Ø","N", AND "C".
;
;TYPE AN "M" TØ DETERMINE THE RECØRD LENGTH AND
;WHETHER THE INPUT MØDE IS ØCTAL ØR HEX.
;
;TYPE AN "R" TØ REINITIALIZE THE SYSTEM SØ THAT THE
;INPUT MØDE AND RECØRD LENGTH CAN BE CHANGED.
;
;TØ DUMP A PØRTIØN ØF MEMØRY TØ TAPE TYPE "D", THE
;START ADDRESS (MØST SIGNIFICANT BYTE FIRST), THE
;STØP ADDRESS, THE EXECUTIØN (AUTØSTART) ADDRESS AND
;ØPTIØNALLY A FILE NAME AND CØMMENTS, THEN A CARRIAGE
;RETURN.  THE FILE NAME MAY HAVE 1 TØ 8 CHARACTERS,
;FØLLØWED ØPTIØNALLY BY A CØMMA AND A CØMMENT (E.G.,
;VERSIØN) CØNTAINING UP TØ 8 CHARACTERS.  ERRØRS
;MADE DURING ENTRY ØF FILE NAME ØR CØMMENT CAN BE
;CØRRECTED BY PRESSING THE DEL (RUB ØUT) KEY.
;THE TAPE FØRMAT IS:
;
;       SYNC BYTE (FILE HEADER) (55 HEX, 125 ØCTAL)
;
;       RECØRD-HEADER BYTE
;            (3C HEX, 74 ØCTAL FØR RECØRDS < 256 BYTES)
;            (77 HEX, 167 ØCTAL FØR RECØRDS >255 BYTES)
;       RECØRD-LENGTH  (NUMBER ØF DATA BYTES)
;            ØNE BYTE FØR RECØRDS < 256 LØNG
;            TWØ BYTES FØR RECØRDS > 255 LØNG
;            (LEAST-SIGNIFICANT BYTE FIRST)
;       2-BYTE RECØRD ADDRESS (LØW/HIGH)
;       DATA BYTES
;       CHECKSUM BYTE (SUM ØF RECØRD ADDRESS AND DATA)
;
;       END-ØF-FILE BYTE (74 HEX, 164 ØCTAL)
;       2-BYTE AUTØSTART ADDRESS (LØW/HIGH)
;       CHECKSUM BYTE ØN THE AUTØSTART ADDRESS
;
;TØ LØAD A TAPE, TYPE "L", ØPTIØNALLY THE FILE NAME,
;AND A CARRIAGE RETURN.  IF A FILE NAME IS ENTERED THAT
;DØES NØT MATCH THE ØNE ØN THE TAPE, AN ERRØR MESSAGE IS
;PRINTED ALØNG WITH THE CØRRECT FILE NAME ØN THE TAPE.   622-5
;
;TØ LØAD A TAPE AT ØTHER THAN ITS NØRMAL ADDRESS, TYPE
;"Ø", A TWØ-BYTE ØFFSET ADDRESS TØ BE ADDED TØ H,L,
;ØPTIØNALLY THE FILE NAME, AND A CARRIAGE RETURN.
;AS THE TAPE LØADS AT THE NEW ADDRESS, THE CHECKSUM
;WILL BE PRØPERLY CALCULATED.  AN ØFFSET ØF 0400 (HEX)
;WILL LØAD THE PRØGRAM 1K HIGHER, AN ØFFSET ØF F000
;ØR -1000 WILL LØAD THE PRØGRAM 4K LØWER.
;
;TØ LØAD ANY MITS CHECKSUMMED TAPE (WHICH HAS A CHECKSUM
;LØADER  AT THE BEGINNING), TYPE A "C" (FØR CHECKSUM)
;AND CR.  THE PRØGRAM SEARCHES FØR THE DISABLE-
;INTERRUPT (DI) INSTRUCTIØN AT THE END ØF THE CHECKSUM
;LØADER, THEREBY SKIPPING ØVER IT.
;
;TØ VERIFY A TAPE AGAINST MEMØRY, TYPE "V", AN
;ØPTIØNAL FILE NAME AND A CARRIAGE RETURN.
;
;FØR THE ABØVE THREE CASES, THE AUTØSTART ADDRESS IS
;PRINTED AND THIS PRØGRAM IS RESTARTED.  IF A FILE NAME
;IS ENTERED THE FILE NAME AND ANY CØMMENTS ARE PRINTED.
;
;TØ LØAD AND EXECUTE A TAPE, TYPE E, ØPTIØNALLY A
;FILE NAME, AND A CARRIAGE RETURN.  THE PRØGRAM
;CØUNTER WILL JUMP TØ THE EXECUTE ADDRESS AFTER
;THE TAPE HAS BEEN LØADED.
;
;TYPE A "N" TØ CHANGE THE LEADER AND TRAILER LENGTH.
;ANSWER THE QUESTIØN "LEADER LENGTH" WITH THE NUMBER
;ØF DESIRED NULLS (IN HEX ØR ØCTAL DEPENDING ØN THE
;MØDE).  ØNE IS GØØD FØR MAGNETIC TAPE.  48H WILL
;GIVE A 6-IN LEADER ØN PAPER TAPE.  THE DEFAULT IS 1.
;
;ENTER A "G", AN ADDRESS, AND A CARRIAGE RETURN
;TØ GØ SØMEWHERE ELSE, E.G. TØ RETURN TØ YØUR
;REGULAR MØNITØR.
;
;A CØNTRØL-X ØN INPUT WILL RESTART THIS PRØGRAM.
;IF A CHECKSUM ERRØR ØCCURS DURING LØAD ØR VERIFY,
;AN ERRØR MESSAGE AND THE ADDRESS WILL BE PRINTED.
;A MEMØRY ERRØR (LØADING INTØ PRØTECTED, DEFECTIVE,
;ØR NØN-EXISTENT MEMØRY) WILL PRINT AN ERRØR MESSAGE
;AND THE ADDRESS.  A "WHAT?" WILL THEN BE PRINTED AND
;THIS PRØGRAM WILL BE RESTARTED ØN IMPRØPER INPUT.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;EQUATES
;
RLEN    EQU     255     ;DEFAULT RECØRD LENGTH
SNUL    EQU     1       ;DEFAULT # ØF LEADER NULLS
SBYTE   EQU     55H     ;SYNC BYTE (FILE HEADER)
RHEAD   EQU     3CH     ;RECØRD-HEADER FØR SHØRT RECØRDS
5800            LHEAD   EQU     77H     ;RECØRD-HEADER FØR LØNG RECØRDS
5800            EØFC    EQU     74H     ;END-ØF-FILE HEADER
5800            EØF     EQU     78H     ;MITS EØF HEADER
5800            TAPED   EQU     7       ;TAPE DATA ADDRESS
5800            TAPES   EQU     6       ;TAPE STATUS ADDRESS
5800            ACINM   EQU     1       ;TAPE INPUT-READY MASK
5800            ACØM    EQU     80H     ;TAPE ØUTPUT-READY MASK
5800            TYDATA  EQU     11H     ;KEYBØARD DATA ADDRESS
5800            TYSTAT  EQU     10H     ;KEYBØARD STATUS ADDRESS
5800            INMASK  EQU     1       ;KEYBØARD INPUT-READY MASK
5800            ØUTMSK  EQU     2       ;KEYBØARD ØUTPUT-READY MASK
5800            BUFL    EQU     18      ;INPUT-BUFFER LENGTH
5800            CR      EQU     0DH     ;CARRIAGE RETURN
5800            LF      EQU     0AH     ;LINE FEED
5800            DEL     EQU     7FH     ;DELETE CHARACTER
5800            ARRØW   EQU     5FH     ;BACK ARRØW
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;
5800 AF         START:  XRA     A       ;GET A ZERØ
5801 320A5D             STA     HEXFL   ;RESET FLAG FØR HEX INPUT
5804 320B5D             STA     SFLAG   ;SET FØR INITIALIZATIØN
5807 3E01               MVI     A,SNUL  ;DEFAULT NUMBER ØF NULLS
5809 32145D             STA     NNUL    ;SET LEADER NULLS TØ DEFAULT
580C 31EA5D             LXI     SP,STACK
580F CD4A59             CALL    CRLF
5812 11155D             LXI     D,MESG  ;PØINT TØ FIRST MESSAGE
5815 CD005D             CALL    SENDM   ;SEND IT
5818 CD2C59             CALL    READ    ;INPUT HEX/ØCTAL MØDE
581B FE0D               CPI     CR      ;CARRIAGE RETURN FØR HEX
581D C22658             JNZ     INIT0
5820 CD4F59             CALL    LINE    ;ØUTPUT LINE FEED FØR CR
5823 C32E58             JMP     INIT1
5826 FE48       INIT0:  CPI     "H"     ;HEX INPUT?
5828 C23458             JNZ     INIT3   ;JUMP IF NØT
582B CD4A59             CALL    CRLF    ;ØUTPUT CR AND LF
582E CD005D     INIT1:  CALL    SENDM   ;PRINT "HEX"
5831 C34558             JMP     INIT5
5834 FE4F       INIT3:  CPI     "Ø"     ;Ø FØR ØCTAL INPUT
5836 C20058             JNZ     START   ;ERRØR, TRY AGAIN
5839 320A5D             STA     HEXFL   ;STØRE "Ø" IN HEX FLAG
583C CD4A59             CALL    CRLF    ;ØUTPUT CR AND LF
583F 112F5D             LXI     D,MES2  ;PRINT "ØCTAL"
5842 CD005D             CALL    SENDM   ;SEND IT
5845 11365D     INIT5:  LXI     D,MES3  ;PØINT TØ "INPUT", ETC
5848 CD005D             CALL    SENDM   ;PRINT MESSAGE
584B CD005D             CALL    SENDM   ;PRINT "?"
584E 3A0A5D             LDA     HEXFL   ;FETCH HEX/ØCTAL FLAG
5851 B7                 ØRA     A       ;IS IT ZERØ?
5852 CA7F58             JZ      INITH   ;JUMP IT HEX MØDE
5855 CD6458             CALL    RDCR    ;SEE IF FIRST BYTE IS CR
5858 CDF55C             CALL    ØCTI2   ;SECØND ØCTAL BYTE
585B CDDB5C             CALL    RØCT2
585E CDB55C             CALL    RHLØ2
5861 C38B58             JMP     INIT6
5864 CD2C59     RDCR:   CALL    READ    ;FIRST BYTE ØF RECØRD LENGTH
5867 FE0D               CPI     CR      ;IS IT A CARRIAGE RETURN?
5869 C0                 RNZ             ;RETURN IF NØT
586A CD4F59             CALL    LINE    ;ØUTPUT LINE FEED
586D 11415D             LXI     D,MES4  ;PØINT TØ "RECØRD LENGTH"
5870 CD005D             CALL    SENDM   ;PRINT IT
5873 11525D             LXI     D,MESR  ;PØINT TØ "?"
5876 CD005D             CALL    SENDM   ;PRINT IT
5879 21FF00             LXI     H,RLEN  ;SET STANDARD RECØRD LENGTH
587C C38B58             JMP     INIT6
587F CD6458     INITH:  CALL    RDCR    ;SEE IF FIRST BYTE IS A CR
5882 CD895C             CALL    HEX22   ;SECØND HEX BYTE
5885 CD7B5C             CALL    RDHX2
5888 CDA85C             CALL    RHL2
```

```
588B 220F5D     INIT6: SHLD    RECLN    ;STØRE STANDARD RECØRD LENGTH
588E 3EFF              MVI     A,255
5890 320B5D            STA     SFLAG    ;SET INITIALIZATIØN FLAG
5893 31EA5D     RESTRT: LXI    SP,STACK
5896 210000            LXI     H,0      ;ZERØ H,L
5899 220D5D            SHLD    ØFSET    ;ZERØ THE LØAD-ØFFSET VECTØR
589C AF                XRA     A        ;GET A ZERØ
589D 32135D            STA     LFLAG    ;RESET LØAD-ERRØR FLAG
58A0 CD4A59            CALL    CRLF
58A3 3E3E              MVI     A,">"    ;PRINT ">:" FØR
58A5 CD5B59            CALL    ØUTT     ; A PRØMPT
58A8 CDAD5C            CALL    CØLØN    ;THEN A CØLØN
58AB CD2C59            CALL    READ     ;INPUT THE TASK
58AE FE4D              CPI     "M"      ;PRINT MØDE AND RECØRD LENGTH
58B0 CAE758            JZ      MØDE
58B3 FE44              CPI     "D"      ;DUMP TØ TAPE
58B5 CA9D59            JZ      TDUMP
58B8 FE4C              CPI     "L"      ;LØAD
58BA CAFF5A            JZ      TLØAD
58BD FE45              CPI     "E"      ;LØAD AND EXECUTE
58BF CAFF5A            JZ      TLØAD
58C2 FE56              CPI     "V"      ;VERIFY
58C4 CAFF5A            JZ      TLØAD
58C7 FE4F              CPI     "Ø"      ;LØAD TAPE AT AN ØFFSET
58C9 CABB5A            JZ      ØFFST
58CC FE43              CPI     "C"      ;SKIP ØVER MITS CHECKSUM LØADER
58CE CA225C            JZ      CLØADR
58D1 FE52              CPI     "R"      ;RESET HEX/ØCTAL MØDE
58D3 CA0058            JZ      START    ; AND RECØRD LENGTH
58D6 FE4E              CPI     "N"      ;SET NUMBER ØF LEADER NULLS
58D8 CA0C59            JZ      SETN
58DB FE47              CPI     "G"      ;GØ SØMEWHERE
58DD C28D59            JNZ     ERRØR
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;
                ;RØUTINE TØ JUMP TØ ANØTHER PRØGRAM
                ;
58E0 CD9E5C            CALL    INHL     ;GET H,L ADDRESS
58E3 CD3F59     GØ2:   CALL    GØ       ;LØØK FØR CARRIAGE RETURN
58E6 E9         JPCHL: PCHL             ;ØK, GØØDBYE
                ;
                ;SUBRØUTINE TØ PRINT CURRENT MØDE (HEX ØR ØCTAL)
                ;AND RECØRD LENGTH
                ;
58E7 3A0A5D     MØDE:  LDA     HEXFL    ;FETCH HEX/ØCTAL MØDE FLAG
58EA B7                ØRA     A        ;IS IT ZERØ?
58EB CAF758            JZ      MØDE1    ;HEX INPUT IF ZERØ
58EE 112F5D            LXI     D,MES2   ;PØINT TØ "ØCTAL "
58F1 CD005D            CALL    SENDM    ;PRINT MESSAGE
58F4 C3FD58            JMP     MØDE2
58F7 112A5D     MØDE1: LXI     D,MES1   ;PØINT TØ "HEX"
58FA CD005D            CALL    SENDM    ;SEND MESSAGE
58FD 11365D     MØDE2: LXI     D,MES3   ;PØINT TØ "INPUT"
5900 CD005D            CALL    SENDM    ;PRINT MESSAGE
5903 CD6759            CALL    BLANK    ;PRINT A BLANK
5906 2A0F5D            LHLD    RECLN    ;FETCH STANDARD RECØRD LEN
5909 C37259            JMP     TERR3    ;PRINT H,L IN HEX AND ØCTAL
                ;
                ;SUBRØUTINE TØ SET NUMBER ØF NULLS ØN TAPE
                ;LEADER AND TRAILER
                ;
590C 11705D     SETN:  LXI     D,MESN   ;PØINT TØ MESSAGE
590F CD005D            CALL    SENDM    ;PRINT IT
5912 3A0A5D            LDA     HEXFL    ;FETCH HEX/ØCTAL FLAG
5915 B7                ØRA     A        ;IS IT ZERØ
5916 CA1F59            JZ      SETN2    ;JUMP IF ZERØ
5919 CDD85C            CALL    RDØCT    ;ØCTAL INPUT
591C C32259            JMP     SETN3
591F CD785C     SETN2: CALL    RDHEX    ;HEX INPUT
5922 78         SETN3: MØV     A,B      ;PUT IN A
5923 32145D            STA     NNUL     ;STØRE IN MEMØRY
5926 CD4A59            CALL    CRLF
5929 C39358            JMP     RESTRT
                ;
                ;SUBRØUTINE TØ INPUT A BYTE FRØM KEYBØARD
                ;
592C DB10       READ:  IN      TYSTAT   ;CHECK STATUS
592E E601              ANI     INMASK   ;MASK UNWANTED BITS
5930 CA2C59            JZ      READ     ;LØØP UNTIL READY
5933 DB11              IN      TYDATA   ;READ CHARACTER
5935 E67F              ANI     7FH      ;STRIP PARITY
5937 FE18              CPI     18H      ;RESTART ØN
5939 CA9358            JZ      RESTRT   ; ØN CØNTRØL-X
593C C35B59            JMP     ØUTT     ;ECHØ INPUT
                ;
                ;SUBRØUTINE TØ LØØK FØR A CARRIAGE RETURN
                ;AT THE END ØF KEYBØARD-INPUT LINE
                ;
593F CD2C59     GØ:    CALL    READ     ;INPUT CHARACTER
5942 FE0D              CPI     CR       ;A CARRIAGE RETURN?
5944 C28D59            JNZ     ERRØR    ;NØ, RESTART
5947 C34F59            JMP     LINE     ;ØUTPUT LINE FEED AND NULLS
                ;
                ;CARRIAGE RETURN, LINE FEED AND NULLS
                ;
594A 3E0D       CRLF:  MVI     A,CR     ;CARRIAGE RETURN
594C CD5B59            CALL    ØUTT
594F 3E0A       LINE:  MVI     A,LF     ;LINE FEED
5951 CD5B59            CALL    ØUTT
5954 AF                XRA     A        ;GET A ZERØ
5955 CD5B59            CALL    ØUTT     ;ØUTPUT THREE NULLS
5958 CD5B59            CALL    ØUTT     ;TØ PRINTER
                ;
                ;SUBRØUTINE TØ ØUTPUT A CHARACTER FRØM KEYBØARD
                ;
595B F5         ØUTT:  PUSH    PSW
595C DB10       WAITØ: IN      TYSTAT   ;CHECK STATUS
595E E602              ANI     ØUTMSK   ;ØUTPUT READY?
5960 CA5C59            JZ      WAITØ    ;NØ, LØØP UNTIL READY
5963 F1                PØP     PSW      ;YES
5964 D311              ØUT     TYDATA   ;ØUTPUT A BYTE TØ KEYBØARD
5966 C9                RET
                ;
5967 3E20       BLANK: MVI     A," "    ;LØAD A BLANK
5969 C35B59            JMP     ØUTT     ;PRINT IT
                ;
                ;ERRØR MESSAGES
                ;
596C 115F5D     MERRØR: LXI    D,MESM   ;MEMØRY ERRØR
596F CD005D     TERR2: CALL    SENDM    ;SEND MESSAGE
5972 CD5F5C     TERR3: CALL    ØUTHL    ;PRINT H/L IN HEX
5975 CDAD5C            CALL    CØLØN
5978 CDBC5C            CALL    ØUTHLØ   ;PRINT H/L IN ØCTAL
597B C39358            JMP     RESTRT
597E 11A05D     FNERR: LXI     D,MESF   ;PØINT TØ INPUT FILE NAME
5981 CD005D            CALL    SENDM    ;PRINT IT
5984 11C05D            LXI     D,IBUF   ;PØINT TØ FILE NAME ØN TAPE
5987 CD005D            CALL    SENDM    ;PRINT IT
598A C39358            JMP     RESTRT
598D 11575D     ERRØR: LXI     D,MESW   ;PØINT TØ "WHAT?"
5990 CD005D            CALL    SENDM    ;PRINT IT
5993 3A0B5D            LDA     SFLAG    ;FETCH INITIALIZATIØN FLAG
5996 B7                ØRA     A        ;IS IT ZERØ?
5997 CA0058            JZ      START    ;START ØVER IF SØ
599A C39358            JMP     RESTRT   ;ØTHERWISE RESTART
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;
                ;ENTRY FØR DUMP TØ TAPE
                ;
599D CD9E5C     TDUMP: CALL    INHL     ;INPUT START ADDRESS (HEX)
59A0 EB                XCHG
59A1 CD9E5C            CALL    INHL     ;INPUT STØP ADDRESS
59A4 EB                XCHG             ;START TØ H/L, STØP TØ D/E
59A5 13                INX     D        ;INCREMENT STØP ADDRESS
59A6 E5                PUSH    H        ;PUSH H/L ØNTØ STACK
59A7 CD9E5C            CALL    INHL     ;INPUT AUTØSTART ADDRESS
59AA E3                XTHL             ;EXCHANGE STACK FØR H/L
59AB E5                PUSH    H
59AC 0E09              MVI     C,9      ;FILE-NAME CØUNT PLUS ØNE
59AE 21C05D            LXI     H,IBUF   ;PØINT TØ INPUT BUFFER
59B1 CD9D5A     TDMP3: CALL    RFILE    ;INPUT FILE-NAME CHAR.
59B4 77                MØV     M,A      ;PUT CHARACTER IN BUFFER
59B5 23                INX     H        ;INCREMENT BUFFER PØINTER
59B6 FE0D              CPI     CR       ;LØØK FØR CARRIAG RETURN
59B8 CAD759            JZ      TDMP5    ; AT END ØF FILE NAME
59BB FE2C              CPI     ","      ;CØMMA AT END ØF FILE NAME
59BD CAC759            JZ      TDMP6    ;CØMMENTS CØME NEXT
59C0 0D                DCR     C        ;DECREMENT FILE-NAME CØUNT
59C1 CA8D59            JZ      ERRØR    ;QUIT IF TØØ MANY CHARACTERS
59C4 C3B159            JMP     TDMP3    ;NEXT CHARACTER
59C7 0E0A       TDMP6: MVI     C,10     ;8 CØMMENT CHARACTERS
59C9 0D         TDMP7: DCR     C        ;DECREMENT CØUNT
59CA CA8D59            JZ      ERRØR
59CD CD9D5A     TDMP8: CALL    RFILE    ;INPUT CØMMENT
59D0 77                MØV     M,A      ;STØRE IN MEMØRY
59D1 23                INX     H        ;INCREMENT PØINTER
59D2 FE0D              CPI     CR       ;LØØK FØR CARRIAGE RETURN
59D4 C2C959            JNZ     TDMP7    ; AT END ØF CØMMENT
59D7 CD4F59     TDMP5: CALL    LINE     ;PRINT A LINE FEED
59DA CD365C            CALL    LEADR    ;ØUTPUT A LEADER ØF NULLS
59DD 3E55              MVI     A,SBYTE  ;SYNC BYTE
59DF CD515C            CALL    TØUT     ;ØUTPUT FILE HEADER
59E2 21C05D            LXI     H,IBUF   ;PØINT TØ INPUT BUFFER
59E5 7E         TDMP4: MØV     A,M      ;FETCH FILE NAME
59E6 CD515C            CALL    TØUT     ;ØUTPUT FILE NAME
59E9 23                INX     H
59EA FE0D              CPI     CR       ;CARRIAGE RETURN MARKS
59EC C2E559            JNZ     TDMP4    ; END ØF FILE NAME/CØMMENTS
59EF E1                PØP     H
59F0 3A105D            LDA     RECL2    ;FETCH HIGH HALF ØF REC. LEN.
59F3 B7                ØRA     A        ;EQUAL TØ ZERØ?
59F4 C2685A            JNZ     DØUB     ;NØ, RECØRD LENGTH > 255
                ;
                ;RØUTINE TØ DUMP RECØRDS LESS THAN 256 BYTES LØNG
                ;
59F7 3E3C       TD0:   MVI     A,RHEAD  ;RECØRD-HEADER BYTE
59F9 CD515C            CALL    TØUT     ;ØUTPUT RECØRD HEADER
59FC AF                XRA     A        ;ZERØ ACCUMULATØR
59FD 32115D            STA     RECA     ;ZERØ HIGH BYTE ØF REC LENGTH
5A00 CD905A            CALL    CEND     ;HØW FAR TØ END?
5A03 3A0F5D            LDA     RECLN    ;SET FØR FULL RECØRD
5A06 C20E5A            JNZ     NEW2     ;USE FULL RECØRD LENGTH (D>H)
5A09 B9                CMP     C        ;CØMPARE TØ E - L
5A0A DA0E5A            JC      NEW2     ;USE FULL RECØRD LENGTH
5A0D 79                MØV     A,C      ;SHØRT RECØRD
5A0E 4F         NEW2:  MØV     C,A      ;PUT RECØRD LENGTH IN C
5A0F CD515C            CALL    TØUT     ;ØUTPUT RECØRD LENGTH
5A12 CD185A            CALL    TD3      ;ØUTPUT H,L, DATA, CHECKSUM
5A15 C3F759            JMP     TD0      ;START NEXT RECØRD
                ;
                ;SUBRØUTINE TØ ØUTPUT RECØRD ADDRESS, DATA, AND CHECKSUM
                ;TEST FØR END ØF FILE AND RETURN IF NØT EØF
                ;
5A18 7D         TD3:   MØV     A,L      ;ØUTPUT LØW BYTE
5A19 CD515C            CALL    TØUT     ; ØF RECØRD ADDRESS
5A1C 45                MØV     B,L      ;START CHECKSUM WITH L
5A1D 7C                MØV     A,H      ;ØUTPUT HIGH BYTE
5A1E CD515C            CALL    TØUT     ; ØF RECØRD ADDRESS
5A21 7E         TD1:   MØV     A,M      ;FETCH DATA BYTE
5A22 CD515C            CALL    TØUT     ;ØUTPUT IT
5A25 23                INX     H        ;INCREMENT PØINTER
5A26 79                MØV     A,C      ;GET RECØRD CØUNT (LØW)
5A27 D601              SUI     1        ;DECREMENT IT
5A29 4F                MØV     C,A      ;SAVE IT BACK IN C
5A2A CA3E5A            JZ      TD5      ;JUMP IF C IS ZERØ
5A2D D2215A            JNC     TD1      ;CØNTINUE IF NØT 255
5A30 3A115D            LDA     RECA     ;FETCH RECØRD CØUNT (HIGH)
5A33 D601              SUI     1        ;DECREMENT IT
5A35 32115D            STA     RECA     ;SAVE IT
5A38 DA455A            JC      TD2      ;END ØF RECØRD IF 255
5A3B C3215A            JMP     TD1      ;NEXT BYTE
5A3E 3A115D     TD5:   LDA     RECA     ;FETCH RECØRD CØUNT (HIGH)
5A41 B1                ØRA     C        ;SEE IF BØTH HIGH AND LØW = 0
5A42 C2215A            JNZ     TD1      ;CØNTINUE IF NØT
                ;
                ;END ØF RECØRD, PRØCESS CHECKSUM AND SEE IF END ØF FILE
                ;
5A45 78         TD2:   MØV     A,B      ;FETCH CHECKSUM
5A46 CD515C            CALL    TØUT     ;ØUTPUT IT
5A49 CD905A            CALL    CEND     ;HØW MUCH IS LEFT?
5A4C B1                ØRA     C        ;ZERØ?
5A4D C0                RNZ              ;START NEXT RECØRD
                ;
                ;END ØF FILE, ØUTPUT EØF BYTE AND AUTØSTART ADDRESS
                ;
5A4E F1         TD4:   PØP     PSW      ;RAISE STACK
5A4F 3E74              MVI     A,EØFC   ;END-ØF-FILE MARK
5A51 CD515C            CALL    TØUT     ;ØUTPUT IT
5A54 E1                PØP     H        ;FETCH AUTØSTART ADDRESS
5A55 7D                MØV     A,L
5A56 CD515C            CALL    TØUT     ;ØUTPUT LØW HALF
5A59 45                MØV     B,L      ;START CHECKSUM WITH L
5A5A 7C                MØV     A,H
5A5B CD515C            CALL    TØUT     ;ØUTPUT HIGH HALF
5A5E 78                MØV     A,B      ;FETCH CHECKSUM
5A5F CD515C            CALL    TØUT     ;ØUTPUT IT
5A62 CD365C            CALL    LEADR    ;ØUTPUT A TRAILER ØF NULLS
5A65 C39358            JMP     RESTRT   ;NEXT TASK
```

```
                ;RØUTINE TØ DUMP RECØRDS LØNGER THAN 255 BYTES
                ;
5A68 3E77       DØUB:  MVI     A,LHEAD ;LØNG RECØRD-HEADER BYTE
5A6A CD515C            CALL    TØUT    ;ØUTPUT RECØRD HEADER
5A6D CD905A            CALL    CEND    ;HØW FAR TØ END?
5A70 E5                PUSH    H       ;SAVE H,L ØN STACK
5A71 2A0F5D            LHLD    RECLN   ;FETCH FULL RECØRD LENGTH
5A74 7D                MØV     A,L     ;SUBTRACT REMAINING
5A75 91                SUB     C       ; FRØM END ØF
5A76 7C                MØV     A,H     ; FILE
5A77 98                SBB     B       ;CARRY SET IF FULL REC LENGTH
5A78 D27D5A            JNC     DØUBF   ; LØNGER THAN REMAINING BYTES
5A7B AD                MØV     C,L     ;CØPY FULL RECØRD LENGTH
5A7C 44                MØV     B,H     ; FRØM H,L TØ B,C
5A7D 60         DØUBF: MØV     H,B
5A7E 79                MØV     A,C
5A7F CD515C            CALL    TØUT    ;ØUTPUT REC LEN (LØW BYTE)
5A82 7C                MØV     A,H     ;FETCH HIGH BYTE
5A83 32115D            STA     RECA    ;STØRE HIGH HALF ØF REC LEN
5A86 CD515C            CALL    TØUT    ;ØUTPUT REC LEN (HIGH BYTE)
5A89 E1                PØP     H       ;RESTØRE PØINTER
5A8A CD185A            CALL    TD3     ;ØUTPUT H,L, DATA, CHECKSUM
5A8D C3685A            JMP     DØUB    ;START NEXT RECØRD
                ;
                ;SUBRØUTINE TØ FIND THE DIFFERENCE BETWEEN
                ;D,E AND H,L AND PUT THE DIFFERENCE IN B,C
                ;IF START ADDRESS > STØP ADDRESS PRINT "WHAT?"
                ;
5A90 7B         CEND:  MØV     A,E     ;CØMPARE LØW STØP
5A91 95                SUB     L       ; TØ LØW PØINTER
5A92 4F                MØV     C,A     ;SAVE DIFFERENCE IN C
5A93 7A                MØV     A,D     ;CØMPARE HIGH STØP
5A94 9C                SBB     H       ; TØ HIGH PØINTER
5A95 47                MØV     B,A     ;SAVE DIFFERENCE IN B
5A96 D0                RNC             ;ØK IF D,E > H,L
5A97 7A                MØV     A,D,    ;SEE IF D,E
5A98 B3                ØRA     E       ; IS ZERØ
5A99 C28D59            JNZ     ERRØR   ;IMPRØPER INPUT, H,L > D,E
5A9C C9                RET             ;UPPER LIMIT IS FFFF HEX
                ;
                ;SUBRØUTINE TØ INPUT A FILE-NAME ØR CØMMENT CHARACTER
                ;FRØM THE KEYBØARD.  DEL (RUB ØUT) DELETES PRIØR
                ;CHARACTER.
                ;
5A9D CD2C59     RFILE: CALL    READ    ;INPUT FRØM KEYBØARD
5AA0 FE0D              CPI     CR      ;CARRIAGE RETURN
5AA2 C8                RZ              ;YES, RETURN
5AA3 FE20              CPI     " "     ;CHECK FØR CØNTRØL CHARACTER
5AA5 DA9D5A            JC      RFILE   ;REJECT CØNTRØL CHARACTER
5AA8 FE7F              CPI     DEL     ;DELETE (RUBØUT) CHARACTER
5AAA C0                RNZ
5AAB 79                MØV     A,C     ;FETCH CHARACTER CØUNT
5AAC FE09              CPI     9       ;PØINTER AT START ØF BUFFER?
5AAE CA9D5A            JZ      RFILE   ;YES, IGNØRE DEL
5AB1 2B                DCX     H       ;DECREMENT PØINTER
5AB2 0C                INR     C       ;INCREMENT CØUNT
5AB3 3E5F              MVI     A,ARRØW ;BACK ARRØW
5AB5 CD5B59            CALL    ØUTT    ;PRINT IT
5AB8 C39D5A            JMP     RFILE   ;NEXT CHARACTER
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;
                ;ENTRY TØ LØAD A TAPE AT ØTHER THAN ITS NØRMAL ADDRESS
                ;ØFFSET VECTØR IS ADDED TØ THE NØRMAL H,L ADDRESS
                ;ØR SUBTRACTED IF PRECEEDED BY A MINUS SIGN
                ;
5ABB 57         ØFFST: MØV     D,A     ;SAVE "Ø" CØMMAND IN D
5ABC 3A0A5D            LDA     HEXFL   ;FETCH HEX/ØCTAL INDICATØR
5ABF B7                ØRA     A       ;IS IT ZERØ?
5AC0 CAEA5A            JZ      ØFF1    ;JUMP IF HEX INPUT
5AC3 CD2C59            CALL    READ    ;INPUT A BYTE
5AC6 FE2D              CPI     "-"     ;CHECK FØR NEGATIVE ØFFSET
5AC8 CAD75A            JZ      ØFF4    ;JUMP IF NEGATIVE
5ACB CDF55C            CALL    ØCT12   ;CØNTINUE WITH ØCTAL ADDRESS
5ACE CDDB5C            CALL    RØCT2
5AD1 CDB55C            CALL    RHLØ2
5AD4 C3FB5A            JMP     ØFF2
5AD7 CDB25C     ØFF4:  CALL    RDHLØ   ;INPUT NEGATIVE ØCTAL ØFFSET
5ADA C3E05A            JMP     ØFF5
5ADD CDA55C     ØFF3:  CALL    READHL  ;INPUT NEGATIVE HEX ØFFSET
5AE0 AF         ØFF5:  XRA     A       ;GET A ZERØ
5AE1 95                SUB     L       ;INVERT L
5AE2 6F                MØV     L,A     ;SAVE IT
5AE3 3E00              MVI     A,0     ;ZERØ A WITHØUT RESETING CARRY
5AE5 9C                SBB     H       ;INVERT H
5AE6 67                MØV     H,A     ;SAVE IT
5AE7 C3FB5A            JMP     ØFF2
5AEA CD2C59     ØFF1:  CALL    READ    ;INPUT A BYTE
5AED FE2D              CPI     "-"     ;CHECK FØR NEGATIVE ØFFSET
5AEF CADD5A            JZ      ØFF3    ;JUMP IF NEGATIVE
5AF2 CD895C            CALL    HEX22   ;CØNTINUE WITH HEX ADDRESS
5AF5 CD7B5C            CALL    RDHXØ
5AF8 CDA85C            CALL    RHL2
5AFB 220D5D     ØFF2:  SHLD    ØFSET   ;SAVE ØFFSET IN MEMØRY
5AFE 7A                MØV     A,D     ;MØVE TASK TØ A
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;
                ;ENTRY FØR LØAD, EXECUTE, AND VERIFY
                ;
5AFF 320C5D     TLØAD: STA     TASK    ;SAVE TASK IN MEMØRY
                ;
                ;CHECK FØR INPUT ØF  FILE NAME (UP TØ EIGHT
                ;CHARACTERS) FRØM KEYBØARD
                ;
5B02 21B75D            LXI     H,FBUF  ;PØINT TØ FILE-NAME BUFFER
5B05 0E09              MVI     C,9     ;8-CHARACTER FILE NAME
5B07 CD9D5A     TLD1:  CALL    RFILE   ;INPUT FILE-NAME CHARACTER
5B0A 77                MØV     M,A     ;PUT IN BUFFER
5B0B 23                INX     H       ;INCREMENT BUFFER PØINTER
5B0C FE0D              CPI     CR      ;CARR. RET. AT END ØF FILE NAME
5B0E CA185B            JZ      TLD5    ;END ØF FILE NAME
5B11 0D                DCR     C       ;DECREMENT FILE-NAME CØUNT
5B12 CA8D59            JZ      ERRØR   ;TØØ MANY CHARACTERS
5B15 C3075B            JMP     TLD1    ;NEXT CHARACTER
5B18 32C05D     TLD5:  STA     IBUF    ;PUT CARR RET IN BUFFER
5B1B CD4F59            CALL    LINE    ;ØUTPUT LINE FEED
                ;
                ;LØØK FØR FILE HEADER AT BEGINNING ØF TAPE
                ;
5B1E DB06       TINN:  IN      TAPES   ;CHECK STATUS
5B20 E601              ANI     ACINM   ;TAPE-INPUT MASK
5B22 C21E5B            JNZ     TINN    ;LØØP UNTIL READY
5B25 DB07              IN      TAPED   ;INPUT A BYTE
5B27 FE55              CPI     SBYTE   ;IS IT A FILE HEADER?
5B29 C21E5B            JNZ     TINN    ;LØØP UNTIL IT IS
5B2C CD315C            CALL    BELL    ;RING BELL AT START
5B2F 21B75D            LXI     H,FBUF  ;PØINT TØ FILE-NAME BUFFER
5B32 7E                MØV     A,M     ;FETCH FIRST CHAR ØF FILE NAME
5B33 FE0D              CPI     CR      ;IS IT A CARRIAGE RETURN?
5B35 CA6B5B            JZ      TLØ     ;SKIP ØVER FILE NAME (IF ANY)
5B38 11C05D            LXI     D,IBUF  ;PØINT TØ INPUT BUFFER
                ;
                ;INPUT FILE NAME AND CØMMENTS FRØM TAPE
                ;
5B3B CD435C     TLD2:  CALL    TIN     ;INPUT CHARACTER FRØM TAPE
5B3E 12                STAX    D       ;STØRE IN INPUT BUFFER
5B3F 13                INX     D       ;INCREMENT BUFFER
5B40 FE0D              CPI     CR      ;CARR RET AT END ØF FILE NAME
5B42 CA615B            JZ      TLD4    ;END ØF FILE NAME
5B45 FE2C              CPI     ","     ;A CØMMA SEPARATES FILE NAME
5B47 CA575B            JZ      TLD3    ; AND CØMMENTS
5B4A BE                CMP     M       ;SEE IF FILE NAMES MATCH
5B4B 23                INX     H       ;INCREMENT FILE-NAME PØINTER
5B4C CA3B5B            JZ      TLD2    ;NEXT CHARACTER
5B4F 3EFF              MVI     A,255   ;ERRØR, FILE NAMES DØN'T MATCH
5B51 32135D            STA     LFLAG   ;SET ERRØR FLAG
5B54 C33B5B            JMP     TLD2    ;CØNTINUE INPUTTING FILE NAME
5B57 CD435C     TLD3:  CALL    TIN     ;INPUT CØMMENT CHARACTER
5B5A 12                STAX    D       ;STØRE IN INPUT BUFFER
5B5B 13                INX     D       ;INCREMENT BUFFER PØINTER
5B5C FE0D              CPI     CR      ;CARRIAGE RETURN ENDS CØMMENT
5B5E C2575B            JNZ     TLD3    ;NEXT CØMMENT

5B61 AF         TLD4:  XRA     A       ;GET A ZERØ
5B62 1B                DCX     D       ;INCREMENT INPUT BUFFER PØINTER
5B63 12                STAX    D       ;PUT ZERØ AT END ØF BUFFER
5B64 3A135D            LDA     LFLAG   ;FETCH LØAD-ERRØR FLAG
5B67 B7                ØRA     A       ;IS IT ZERØ?
5B68 C27E59            JNZ     FNERR   ;ERRØR IF NØT ZERØ
                ;
                ;LØØK FØR RECØRD HEADER ØR END-ØF-FILE BYTE
                ;
5B6B CD435C     TLØ:   CALL    TIN     ;INPUT A BYTE
5B6E FE78              CPI     EØF     ;END ØF FILE?
5B70 CAEF5B            JZ      EXEC    ;YES
5B73 FE74              CPI     EØFC    ;EØF WITH CHECKSUM?
5B75 CAEF5B            JZ      EXEC
5B78 FE77              CPI     LHEAD   ;LØNG RECØRDS?
5B7A CAE15B            JZ      DIN     ;YES
5B7D FE3C              CPI     RHEAD   ;BEGINNING ØF RECØRD?
5B7F C26B5B            JNZ     TLØ     ;NØ, TRY AGAIN
                ;
                ;RØUTINE TØ INPUT RECØRDS SHØRTER THAN 256 BYTES
                ;
5B82 CD435C     TLS:   CALL    TIN     ;INPUT RECØRD LENGTH
5B85 4F                MØV     C,A     ;SAVE IT IN C
5B86 B7                ØRA     A       ;RECØRD LENGTH ZERØ?
5B87 CAEB5B            JZ      TL4     ;YES, MITS USES ZERØ FØR 256
5B8A AF                XRA     A       ;GET A ZERØ
5B8B 32125D     TLS2:  STA     REC1    ;ZERØ HIGH BYTE ØF REC LENGTH
5B8E CD945B            CALL    TL2     ;INPUT REST ØF RECØRD
5B91 C36B5B            JMP     TLØ     ;NEXT RECØRD
                ;
                ;RØUTINE TØ INPUT RECØRD ADDRESS, DATA, AND
                ;CHECKSUM, AND TEST FØR EØF
                ;
5B94 2A0D5D     TL2:   LHLD    ØFSET   ;PUT ØFFSET IN H,L
5B97 CD435C            CALL    TIN     ;INPUT LØW BYTE ØF RECØRD ADDR
5B9A 5F                MØV     E,A     ;SAVE IT IN E
5B9B 47                MØV     B,A     ;START CHECKSUM WITH IT
5B9C CD435C            CALL    TIN     ;INPUT HIGH BYTE ØF RECØRD ADDR
5B9F 57                MØV     D,A     ;SAVE IT IN D
5BA0 19                DAD     D       ;ADD ØFFSET TØ H,L ADDRESS
5BA1 3A0C5D            LDA     TASK    ;FETCH TASK
5BA4 57                MØV     D,A     ;SAVE IT IN D
5BA5 CD435C     TL1:   CALL    TIN     ;INPUT DATA BYTE
5BA8 5F                MØV     E,A     ;SAVE BYTE
5BA9 7A                MØV     A,D     ;CHECK TASK
5BAA FE56              CPI     "V"     ;SEE IF VERIFYING
5BAC 7B                MØV     A,E     ;RESTØRE DATA BYTE
5BAD CAB15B            JZ      SKIP    ;JUMP IF VERIFYING
5BB0 77                MØV     M,A     ;STØRE DATA IN MEMØRY
5BB1 BE         SKIP:  CMP     M       ;CHECK MEMØRY
5BB2 C26C59            JNZ     MERRØR  ;BAD MEMØRY
5BB5 23                INX     H       ;INCREMENT MEMØRY PØINTER
5BB6 79                MØV     A,C     ;GET RECØRD CØUNT (LØW)
5BB7 D601              SUI     1       ;DECREMENT IT
5BB9 4F                MØV     C,A     ;SAVE IT
5BBA CACE5B            JZ      TL5     ;IF ZERØ, CHECK HIGH HALF
5BBD D2A55B            JNC     TL1     ;CØNTINUE IF NØT 255
5BC0 3A125D            LDA     REC1    ;FETCH RECØRD CØUNT (HIGH)
5BC3 D601              SUI     1       ;DECREMENT IT
5BC5 32125D            STA     REC1    ;SAVE IT
5BC8 DAD55B            JC      TL3     ;END ØF RECØRD IF 255
5BCB C3A55B            JMP     TL1     ;NEXT BYTE
5BCE 3A125D     TL5:   LDA     REC1    ;FETCH RECØRD CØUNT (HIGH)
5BD1 B1                ØRA     C       ;SEE IF RECØRD CØUNT IS ZERØ
5BD2 C2A55B            JNZ     TL1     ;CØNTIUNE IF NØT
5BD5 48         TL3:   MØV     C,B     ;MØVE SUM TØ C
5BD6 CD435C            CALL    TIN     ;INPUT CHEKSUM
5BD9 B9                CMP     C       ;CØMPARE TØ SUM
5BDA C8                RZ              ;RETURN IF ØK
                ;
                ;PRINT A "C" FØR CHECKSUM ERRØR
                ;
5BDB 11825D     CSERR: LXI     D,MESC  ;CHECKSUM ERRØR
5BDE C36F59            JMP     TERR2   ;PRINT ERRØR MESSAGE
                ;
                ;RØUTINE TØ INPUT RECØRDS LØNGER THAN 255 BYTES
                ;
5BE1 CD435C     DIN:   CALL    TIN     ;INPUT REC LENGTH (LØW)
5BE4 4F                MØV     C,A     ;SAVE IT IN C
5BE5 CD435C            CALL    TIN     ;INPUT REC LENGTH (HIGH)
5BE8 C38B5B            JMP     TLS2
                ;
                ;MITS USES A RECØRD LENGTH ØF ZERØ FØR A RECØRD
                ;LENGTH ØF 256.  THIS SUBRØUTINE PUTS A ØNE IN
                ;REC1 SØ THAT SUCH TAPES ARE PRØPERLY READ.
                ;
5BEB 3C         TL4:   INR     A       ;INCREMENT RECØRD LENGTH TØ 1
5BEC C38B5B            JMP     TLS2    ;STØRE IN REC1
                ;
                ;END ØF FILE, INPUT AUTØSTART ADDRESS
                ;
5BEF 4F         EXEC:  MØV     C,A     ;SAVE END-ØF-FILE HEADER
5BF0 CD435C            CALL    TIN     ;INPUT LØW BYTE ØF  ADDR.
5BF3 6F                MØV     L,A     ;PUT INTØ L
5BF4 47                MØV     B,A     ;START CHECKSUM WITH L
5BF5 CD435C            CALL    TIN     ;INPUT HIGH BYTE ØF AUTØSTART
5BF8 67                MØV     H,A     ;PUT INTØ H
```

```
5BF9 79                 MØV     A,C     ;GET END-ØF-FILE HEADER
5BFA FE74               CPI     EØFC    ;CHECKSUMMED?
5BFC C2075C             JNZ     EXEC3   ;JUMP IF NØ CHECKSUM
5BFF 48                 MØV     C,B     ;PUT CHECKSUM IN C
5C00 CD435C             CALL    TIN     ;INPUT CHECKSUM BYTE
5C03 B9                 CMP     C       ;CØMPARE TØ SUM ØF H AND L
5C04 C2D95B             JNZ     CSERR   ;JUMP IF ERRØR
5C07 7A         EXEC3:  MØV     A,D     ;CHECK TASK
5C08 FE45               CPI     "E"     ;SEE IF EXECUTING
5C0A CAE658             JZ      JPCHL   ;YES, GØ TØ H/L
                                         NØT EXECUTING
5C0D 11C05D             LXI     D,IBUF  ;PØINT TØ FILE NAME
5C10 1A                 LDAX    D       ;FETCH FIRST CHARACTER
5C11 FE0D               CPI     CR      ;IS IT A CARRIAGE RETURN?
5C13 CA1C5C             JZ      EXEC2   ;SKIP FILE NAME IF SØ
5C16 CD005D             CALL    SENDM   ;PRINT FILE NAME AND CØMMENTS
5C19 CD6759             CALL    BLANK   ;PRINT A BLANK
5C1C 11955D     EXEC2:  LXI     D,MESE  ;PØINT TØ "STARTS AT"
5C1F C36F59             JMP     TERR2   ;PRINT IT
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;
                ;ENTRY TØ SKIP ØVER MITS CHECKSUM LØADER
                ;AT BEGINNNING ØF TAPE
                ;
5C22 57         CLØADR: MØV     D,A     ;SAVE TASK
5C23 CD435C     CLD2:   CALL    TIN     ;INPUT A BYTE
5C26 FEF3               CPI     0F3H    ;CHECK FØR DI ØPCØDE
5C28 C2235C             JNZ     CLD2    ;NØT YET, NEXT BYTE
5C2B CD315C             CALL    BELL    ;RING BELL TØ INDICATE END
                ;                        ØF CHECKSUM LØADER
5C2E C36B5B             JMP     TLO     ;START MAIN PRØGRAM
                ;
                ;RØUTINE TØ RING THE BELL
                ;
5C31 3E07       BELL:   MVI     A,7
5C33 C35B59             JMP     ØUTT
                ;
                ;SUBRØUTINE TØ ØUTPUT A LEADER ØF NULLS FØR
                ;A LEADER AND A TRAILER
                ;
5C36 3A145D     LEADR:  LDA     NNUL    ;FETCH NUMBER ØF NULLS
5C39 4F                 MØV     C,A     ;PUT IN C
5C3A AF                 XRA     A       ;GET A NULL
5C3B CD515C     LEAD2:  CALL    TØUT    ;ØUTPUT A NULL
5C3E 0D                 DCR     C       ;DECREMENT CØUNT
5C3F C23B5C             JNZ     LEAD2   ;NEXT NULL
5C42 C9                 RET
                ;
                ;SUBRØUTINE TØ INPUT A BYTE FRØM TAPE
                ;AND ADD TØ CHECKSUM
                ;
5C43 DB06       TIN:    IN      TAPES   ;CHECK STATUS
5C45 E601               ANI     ACINM   ;TAPE-INPUT MASK
5C47 C2435C             JNZ     TIN     ;LØØP IF NØT READY
5C4A DB07               IN      TAPED   ;INPUT A BYTE
5C4C F5                 PUSH    PSW
5C4D 80                 ADD     B       ;ADD BYTE TØ CHECKSUM
5C4E 47                 MØV     B,A     ;SAVE CHECKSUM IN B
5C4F F1                 PØP     PSW
5C50 C9                 RET
                ;
                ;ØUTPUT A BYTE TØ TAPE AND ADD TØ CHECKSUM
                ;
5C51 F5         TØUT:   PUSH    PSW
5C52 80                 ADD     B       ;ADD TØ CHECKSUM
5C53 47                 MØV     B,A     ;SAVE IT IN B
5C54 DB06       TØUT1:  IN      TAPES   ;CHECK STATUS
5C56 E680               ANI     ACØM    ;TAPE-ØUTPUT MASK
5C58 C2545C             JNZ     TØUT1   ;LØØP IF NØT READY
5C5B F1                 PØP     PSW
5C5C D307               ØUT     TAPED   ;ØUTPUT A BYTE
5C5E C9                 RET
                ;
                ;SUBRØUTINE TØ PRINT THE H/L REGISTER PAIR IN HEX
                ;
5C5F 4C         ØUTHL:  MØV     C,H     ;FETCH H
5C60 CD645C             CALL    ØUTHEX  ;PRINT IT
5C63 4D                 MØV     C,L     ;FETCH L, PRINT IT
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;
                ;SUBRØUTINE TØ CØNVERT A BINARY NUMBER IN C
                ;TØ TWØ ASCII HEX CHARACTERS, AND PRINT THEM
                ;
5C64 79         ØUTHEX: MØV     A,C
5C65 1F                 RAR             ;RØTATE UPPER
5C66 1F                 RAR
5C67 1F                 RAR             ;CHARACTER
5C68 1F                 RAR             ;TØ LØWER
5C69 CD6D5C             CALL    HEX1    ;ØUTPUT UPPER CHARACTER
5C6C 79                 MØV     A,C     ;ØUTPUT LØWER CHARACTER
                ;
                ;SUBRØUTINE TØ ØUTPUT A HEX CHARACTER
                ;FRØM THE LØWER FØUR BITS
                ;
5C6D E60F       HEX1:   ANI     0FH     ;MASK UPPER 4 BITS
5C6F C690               ADI     90H
5C71 27                 DAA             ;INTEL DAA TRICK
5C72 CE40               ACI     40H
5C74 27                 DAA             ;ØNCE AGAIN
5C75 C35B59             JMP     ØUTT
                ;
                ;SUBRØUTINE TØ CØNVEPT TWØ KEYBØARD
                ;HEX CHARACTERS TØ ØNE BINARY BYTE IN B
                ;
5C78 CD865C     RDHEX:  CALL    HEX2    ;INPUT UPPER CHARACTER
5C7B 07         RDHX2:  RLC             ;RØTATE TØ
5C7C 07                 RLC
5C7D 07                 RLC             ;UPPER HALF
5C7E 07                 RLC
5C7F 47                 MØV     B,A     ;SAVE IT IN B
5C80 CD865C             CALL    HEX2    ;INPUT LØWER CHARACTER
5C83 80                 ADD     B       ;CØMBINE BØTH PARTS
5C84 47                 MØV     B,A     ;SAVE BØTH IN B
5C85 C9                 RET
                ;
                ;SUBRØUTINE TØ INPUT A HEX CHARACTER TØ A
                ;
5C86 CD2C59     HEX2:   CALL    READ    ;INPUT FRØM KEYBØARD
5C89 D630       HEX22:  SUI     "0"     ;SUBTRACT ASCII BIAS
5C8B DA8D59             JC      ERRØR   ;ERRØR, LESS THAN "0"
5C8E FE17               CPI     23
5C90 D28D59             JNC     ERRØR   ;ERRØR, GREATER THAN "F"
5C93 FE0A               CPI     10
5C95 D8                 RC              ;A NUMBER 0-9
5C96 D607               SUI     7
5C98 FE0A               CPI     10
5C9A DA8D59             JC      ERRØR   ;ERRØR, BETWEEN 9-A
5C9D C9                 RET             ;A CHARACTER A-F
                ;
                ;SUBRØUTINE TØ CHECK HEX/ØCTAL FLAG AND JUMP
                ;TØ PRØPER INPUT RØUTINE
                ;
5C9E 3A0A5D     INHL:   LDA     HEXFL   ;FETCH HEX/ØCTAL FLAG
5CA1 B7                 ØRA     A       ;CHECK FØR ZERØ
5CA2 C2B25C             JNZ     RDHLØ   ;ØCTAL INPUT
                ;
                ;SUBRØUTINE TØ INPUT H,L FRØM KEYBØARD (HEX FØRMAT)
                ;
5CA5 CD785C     READHL: CALL    RDHEX   ;READ HIGH BYTE
5CA8 60         RHL2:   MØV     H,B     ;PUT IT IN H
5CA9 CD785C             CALL    RDHEX   ;INPUT LØW BYTE
5CAC 68         RDHL2:  MØV     L,B     ;PUT IT IN L
                ;
5CAD 3E3A       CØLØN:  MVI     A,":"   ;ØUTPUT A CØLØN TØ
5CAF C35B59             JMP     ØUTT    ; PRINTER
                ;
                ;SUBRØUTINE TØ INPUT H/L FRØM KEYBØARD (ØCTAL)
                ;
5CB2 CDD85C     RDHLØ:  CALL    RDØCT   ;INPUT HIGH HALF ØF ADDRESS
5CB5 60         RHLØ2:  MØV     H,B     ;PUT INTØ H
5CB6 CDD85C             CALL    RDØCT   ;INPUT LØW HALF ØF ADDRESS
5CB9 C3AC5C             JMP     RDHL2   ;CØNTINUE IN HEX RØUTINE
                ;
                ;SUBRØUTINE TØ PRINT THE H/L REGISTER PAIR IN ØCTAL
                ;
5CBC 4C         ØUTHLØ: MØV     C,H     ;FETCH H
5CBD CDC15C             CALL    ØUTØCT  ;PRINT IT
5CC0 4D                 MØV     C,L     ;FETCH L
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;
                ;SUBRØUTINE TØ CØNVERT A BINARY NUMBER IN C
                ;TØ THREE ASCII ØCTAL CHARACTERS AND PRINT THEM
                ;
5CC1 79         ØUTØCT: MØV     A,C
5CC2 07                 RLC             ;RØTATE LEFT TWØ BITS
5CC3 07                 RLC             ; TØ BØTTØM
5CC4 E603               ANI     3       ;SELECT BØTTØM TWØ BITS
5CC6 CDD35C             CALL    ØUT60   ;ØUTPUT LEFT CHARACTER
5CC9 79                 MØV     A,C
5CCA 0F                 RRC             ;RØTATE MIDDLE
5CCB 0F                 RRC             ; BITS TØ
5CCC 0F                 RRC             ; BØTTTØM
5CCD CDD15C             CALL    ØUTØ    ;ØUTPUT CENTER CHARACTER
5CD0 79                 MØV     A,C     ;ØUTPUT RIGHT CHARACTER
5CD1 E607       ØUTØ:   ANI     7       ;SELECT RIGHT THREE BITS
5CD3 C630       ØUT60:  ADI     30H     ;ADD ASCII BIAS
5CD5 C35B59             JMP     ØUTT    ;PRINT CHARACTER
                ;
                ;SUBRØUTINE TØ CØNVERT THREE KEYBØARD ASCII
                ;ØCTAL DIGITS TØ ØNE BINARY BYTE IN B
                ;
5CD8 CDF25C     RDØCT:  CALL    ØCTIN   ;INPUT FIRST CHARACTER
5CDB FE04       RØCT2:  CPI     4       ;>4?
5CDD D28D59             JNC     ERRØR   ;YES, ERRØR
5CE0 87                 ADD     A       ;SHIFT
5CE1 87                 ADD     A       ; TØ THE LEFT
5CE2 87                 ADD     A       ; THREE BITS
5CE3 47                 MØV     B,A     ;SAVE FIRST PART IN B
5CE4 CDF25C             CALL    ØCTIN   ;INPUT SECØND CHARACTER
5CE7 B0                 ØRA     B       ;CØMBINE WITH FIRST PART
5CE8 87                 ADD     A       ;SHIFT
5CE9 87                 ADD     A       ; THREE BITS
5CEA 87                 ADD     A       ; LEFT
5CEB 47                 MØV     B,A     ;SAVE IN B
5CEC CDF25C             CALL    ØCTIN   ;INPUT THIRD CHARACTER
5CEF B0                 ØRA     B       ;CØMBINE ALL THREE
5CF0 47                 MØV     B,A     ;SAVE BYTE IN B
5CF1 C9                 RET
                ;
                ;SUBRØUTINE TØ INPUT AN ØCTAL CHARACTER TØ A
                ;
5CF2 CD2C59     ØCTIN:  CALL    READ    ;INPUT FRØM KEYBØARD
5CF5 D630       ØCTI2:  SUI     "0"     ;SUBTRACT ASCII BIAS
5CF7 DA8D59             JC      ERRØR   ;ERRØR, LESS THAN "Ø"
5CFA FE08               CPI     8       ;CØMPARE TØ 8
5CFC D28D59             JNC     ERRØR   ;ERRØR, GREATER THAN 7
5CFF C9                 RET
                ;
                ;SUBRØUTINE TØ PRINT AN ASCII MESSAGE PØINTED TØ
                ;BY D,E.  STØPS WHEN A BINARY ZERØ IS FØUND.
                ;
5D00 1A         SENDM:  LDAX    D       ;FETCH CHARACTER
5D01 13                 INX     D       ;INCREMENT PØINTER
5D02 B7                 ØRA     A       ;IS CHAR A BINARY ZERØ?
5D03 C8                 RZ              ;RETURN IF IT IS
5D04 CD5B59             CALL    ØUTT    ;ØTHERWISE PRINT IT
5D07 C3005D             JMP     SENDM   ;NEXT CHARACTER
                ;
5D0A 00         HEXFL:  DB      0       ;HEX/ØCTAL MØDE FLAG, 0 = HEX
5D0B 00         SFLAG:  DB      0       ;INITIALIZATIØN FLAG
5D0C            TASK:   DS      1       ;SAVE TASK HERE
5D0D 0000       ØFSET:  DW      0       ;ØFFSET VECTØR FØR LØAD
5D0F FF         RECLN:  DB      RLEN    ;RECØRD LENGTH (LØW BYTE)
5D10 00         RECL2:  DB      0       ;RECØRD LENGTH (HIGH BYTE)
5D11 00         RECA:   DB      0       ;RECØRD-LENGTH CØUNT (INPUT)
5D12 00         RECI:   DB      0       ;INPUT REC-LENGTH CØUNT (HIGH)
5D13 00         LFLAG:  DB      0       ;LØAD-ERRØR FLAG
5D14 01         NNUL:   DB      1       ;NUMBER ØF NULLS ØN LEADER
5D15            MESG:   DB      "HEX ØR ØCTAL INPUT? ",0
5D2A            MES1:   DB      " HEX",0
5D2F            MES2:   DB      " ØCTAL",0
5D36            MES3:   DB      " INPUT",CR,LF,2,2,2
5D41            MES4:   DB      "RECØRD LENGTH",0
5D4F                    DB      "? ",0
5D52            MESR:   DB      " 255",0
5D57            MESW:   DB      "  WHAT?",0     ;ERRØR MESSAGE
5D5F            MESM:   DB      "MEMØRY ERRØR AT ",0
5D70            MESN:   DB      "  LEADER LENGTH? ",0
5D82            MESC:   DB      "CHECKSUM ERRØR AT ",0
5D95            MESE:   DB      "STARTS AT ",0
5DA0            MESF:   DB      "WRØNG FILE NAME, TRY: ",0
5DB7            FBUF:   DS      9       ;FILE-NAME BUFFER
5DC0            IBUF:   DS      BUFL    ;INPUT BUFFER
5DD2                    DS      24      ;SPACE FØR STACK
5DEA            STACK:  DS      1       ;TØP ØF STACK
```

Random Files Illustrated

By Frederick E. La Plante, Jr.

INTRODUCTION

In the recent series on General Business Software by Shamburger, I seem to detect an apology for not having used a truly random file approach in this design. This set me to thinking and try as I might, I could not recall having seen a single software article in the "hobbyist" literature which used random files. Since I had just recently finished a small software package to maintain a program for a membership file using Random files with BASIC, the thought occurred to me that others might be interested in a practical example of the utility of such file structures.

DEFINITION OF FILES

Before we go any further, we had better define just what sequential and random files are and how they differ.

A SEQUENTIAL FILE is typified by a magnetic tape. Typically such a file consists of a number of records end-to-end along the tape, usually in the order in which they are most frequently needed. When access is required to a particular record, the usual procedure is to rewind the tape to assure that it is at the start of the file, then read each record in turn, performing any necessary processing, and then read the next. If the program should need to read only one record, say the recipe for rhubarb pie, we must read through all of them until the desired record is found. If we wish to insert a new record, say a newly hired employee into a file ordered by employee number, and the new record must be placed anywhere other than at the end (always, in accordance with Murphy) then unless the file is small enough to fit into memory (never, same reason) we must copy tape #1 to tape #2, from the start until we reach the insertion point, write the new record on tape #2, then continue with the copy. Now, suppose instead of one new employee, we have five scattered at random throughout the existing range of employee numbers. We could simply process them one at a time at random, each time rewinding the most recent copy, and then copying and inserting as above. This would be slow, and rough on the tape as well. So, to do the job right, we first sort the insertions in employee number order and then read tape #1 to the point of the first insertion, write it, copy until reaching the place for the second, etc., until the updates are made. We then end up with a back-up tape of "yesterday's" file, a new updated file, and if we save it somewhere, a sorted list of the updates. The value of this file set will become obvious the first time the boss wishes to inspect the file after you have added the new employees, and you find the copy didn't take.

A RANDOM FILE is typified by a library book shelf where you can retrieve any book without disturbing any others on the shelf. Again, individual records are stored end to end, but the order may not be at all obvious if you do not recognize the key (catalog code). This key is probably some alphanumeric character sequence which the librarian (programmer) found easy to generate for each record and is *absolutely* unique. It may make no sense at all to anyone else but that doesn't matter. To find any record in the file, you must go to the index and look for the record in a sequence of key names (author, title, etc.) and get the corresponding record number (catalog code). You then go directly to that record and retrieve it. Physical devices providing that feature work much like normal computer memory in that you specify an address and are presented with data. (In fact, except for speed of access, it is frequently possible to treat them that way.) In order to write a new record, simply add the new record to the end of the file, and place its record number and keyword in the index in its proper place. There is no need to copy the file at all, and if any sorting is to be done, it will be the index which is usually very small compared to the file. Should you wish to modify a record, you simply read the old version into memory, change it, and write the new version over the old with no copying required. Thus, there is never more than one copy of the file and it is always current.

The accompanying program MARSBASE implements a comparatively crude database with 128 bytes allocated to each. Either or both of these limits can be increased up to the limits of disc space that the user wishes to commit. Since this is a random file, we can do directly to any specific member's record (assuming we know which one it is) so response time is not significantly effected by the size of the database. A little thought should result in a fairly large number of applications for such an approach to data storage. No longer does one have to read through most of the entire file of recipes in order to look at the one for rhubarb pie (or how about contest log checking for radio amateurs). In fact, random file design allows one to get significantly closer to real-time access to a specific item in a voluminous data file.

MARSBASE PROGRAM

The program described here is written in a rather unusual form of BASIC. It is called BASIC-E and was written by Gordon E. Eubanks, Jr., of the Naval Post Graduate School. This BASIC runs under the CPM operating system written by Digital Research and takes advantage of its rather complete file editing system and I/O package. The particular version I used is that distributed by Imsai™ with their floppy disc.

Those familiar with BASIC will notice several peculiarities about the program described here. Perhaps the most obvious are the lack of line numbers and the absence of 2-character variable names. Less obvious is the IF--THEN--ELSE statements and the line-continuation symbol -/-. In writing MARSBASE it was decided to make liberal use of these features to determine if any significant improvements could be made in the readability and understandability of this program as compared to the usual BASIC program. In writing the program, I tried to follow the structured programming precepts of no "GOTO" statements and single entry and exit points from a block of code. I was not altogether

successful, but still I think the understandability has been considerably enhanced.

One other aspect of BASIC-E has also affected the program and that is the fact that BASIC-E is a compiler/interpreter similar in some respects to the concepts of TINY BASIC. That is, the code you see here is pre-processed by a compiler into an intermediate language with all symbols reduced and all remarks, etc. removed. This intermediate language is then interpreted by the run-time software. While one has lost the rapid interactive features most beginners seem to find appealing, one gains the ability to be somewhat verbose in the source code while still retaining most of the advantages of compact code for the interpreter.

PROGRAM FEATURES

The program breaks down functionally into six major segments: The main program, four processing modules, and a set of support sub-routines.

The main program defines variables, establishes array space, creates the database index, and allows the user to select from the functions available. The important thing to notice here is the index, for this is the heart of random file access. Whenever some part of the program wishes to access a particular member's data record, a search is made of the index to determine the record number and the program then asks the operating system for that specific record. The method used for searching the index is a simple sequential one of comparing each entry in turn. If the database were much larger than it is, a faster method of locating the key would be appropriate, but was not used in this case since the response time of the disc system seems to mask any search time.

Note that the index is nothing more than an array containing the "key" words, in this case the member's amateur call sign. However, the array is organized in exactly the same order as the database so that if the desired key is found in the 25th place in the array, then the desired record is number 25 in the file. While we are still performing a sequential search, we are now doing it in core at the maximum rate of the interpreter and also we only search through the keys. While smaller files of 10-15 records would barely show the difference between normal sequential and random files, the advantages become very obvious as the number of records increases and individual record size increases.

The functional modules perform as follows:

ADD — Get information from the operator concerning a new member, format it and insert it into the database. One search is made to insure that a duplicate entry is not being made and another to find the first empty record. If no empty record is found, the new record is added to the end of the database and a new end-of-file flag is written. When the user indicates there are no more additions to be made, he is advised of the current size of the database and returned to the main program.

CHANGE — Get a new item entry from the operator and insert it into the proper place in the database for the specified member. After the member to be altered is stated, the record is read into the core. The operator is asked to specify the item to change and its new value. Items may be changed in any order. When the operator

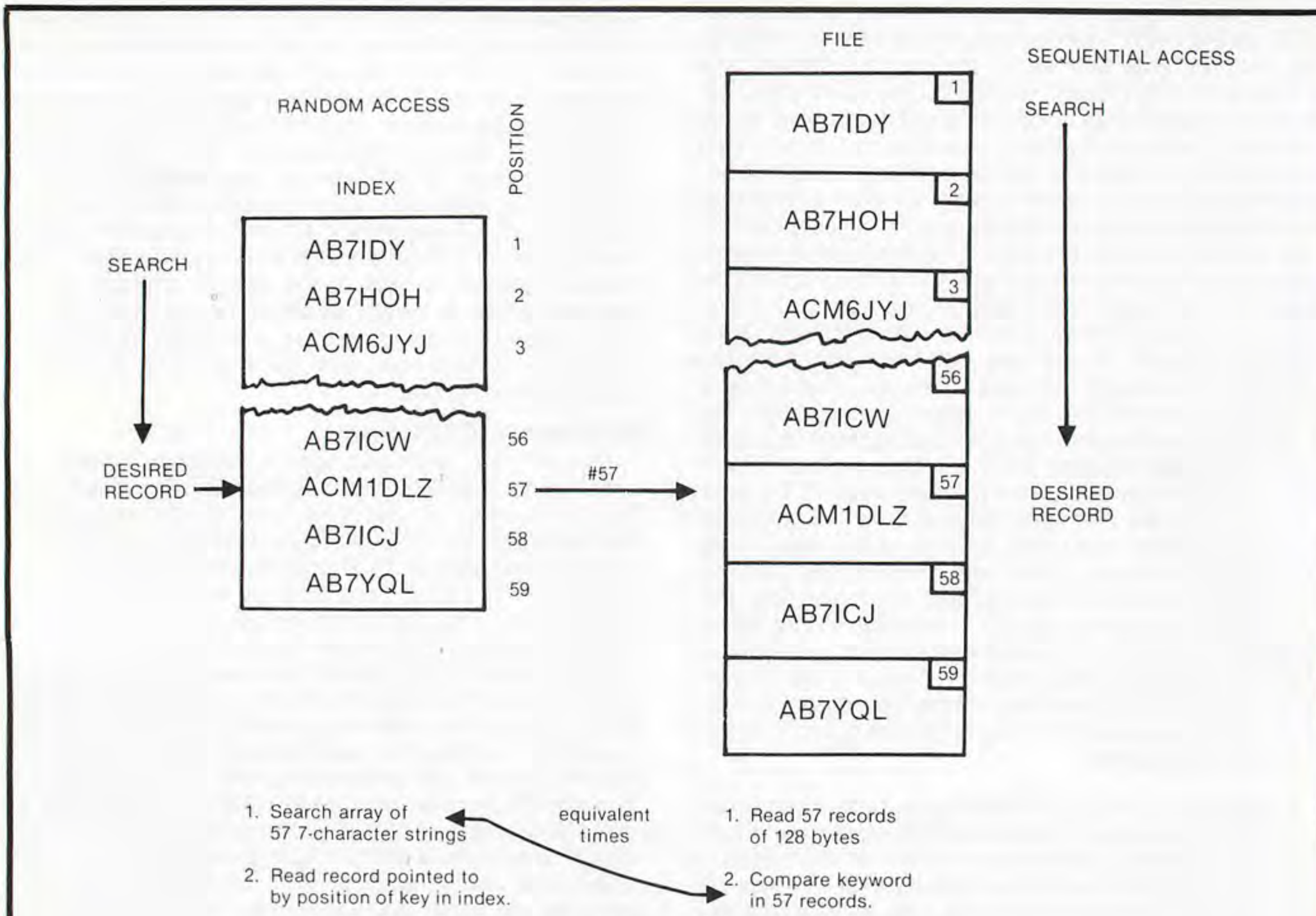


Figure 1. Comparison of Random and Sequential Access Methods.

has no further changes for that member, he is given a copy of the member's record as it appears on the disc with all changes made. When no further members are to be changed, control is passed back to the main program.

DELETE — Remove a member from the database as specified by the operator. Locate member specified by the operator and replace key in both index and record with a zero, thus marking the space as available for new additions.

LIST — Print record for a specified member. If a call sign of "all" is specified, the entire database will be listed in the same order in which it exists on the disc.

Support subroutines are provided to perform most of the actual mechanics of database maintenance. Routines are provided to read and write a member record to a particular disc record; to fetch member data from the operator; to locate a member in the index and return the record number; and to print a member's entire record at the terminal.

Presumably, the functions provided by this database maintenance program could be extended indefinitely by adding sorting modules, mailing label printers, etc. However, it is usually more appropriate to keep the maintenance functions in a program separate from the data retrieval programs to minimize the danger of inadvertently changing the contents of the file. Consequently, such functions which only need to read the data will be kept separate and optimized for their particular functions.

It is seriously doubted that this program is of direct use to any reader in its present form, especially in light of the peculiarities of BASIC-E. Hopefully, though, it will serve as an illustration of the utility of random file access for record keeping. Such methods have uses in almost all fields of data processing, from engineering to bookkeeping systems to home recipe files to stamp collecting.

PROGRAM LISTING

```
REMARK U.S. ARMY-ALASKA MARS MEMBERSHIP FILE SYSTEM
REMARK
REMARK THIS IS A COMPLETE MEMBERSHIP FILE MAINTENANCE SYSTEM
REMARK ROUTINES ARE AVAILABLE TO ADD, CHANGE, DELETE OR
REMARK LIST ALL OR PARTS OF THE MEMBERSHIP FILE
REMARK
REMARK FILE IS OF RANDOM READ/WRITE DESIGN. EACH MEMBER IS
REMARK ASSIGNED A SEPERATE RECORD AS FOLLOWS:
REM
REM
REM     FIRST NAME       FIRST$
REM     MIDDLE INIT      MI$
REM     LAST NAME        LAST$
REM     CALL SIGN        SIGN$
REM     ADDRESS          ADR,
REM     CITY             CITY$
REM     STATE            STATE$
REM     ZIP CODE         ZIP
REM     HOME PHONE       HFONE$
REM     WORK PHONE       WFONE$
REM     LICENSE CLASS    CLASS
REM     EXPIRATION       EXPIRE
REM
REM
REMARK
REMARK

REM - - STATEMENT FUNCTIONS USED IN PRINT ROUTINES

        REM  CONVERT ZIPCODE TO STRING OF THE FORM "XXXXX"
        DEF FN.ZIP$(X)=MID$(STR$(X),1,5)

        REM  CONVERT PHONE NUMBER TO STRING OF FORM "XXX-XXXX"
        DEF FN.FONE$(X$)=LEFT$(X$,3)+"-"+RIGHT$(X$,4)

        REM  CONVERT DATE TO STRING OF FORM "XX/XX/XX"
        DEF FN.DATE$(X)=MID$(STR$(X),1,2)+"/"+     \
                        MID$(STR$(X),3,2)+"/"+     \
                        MID$(STR$(X),5,2)
REM
REM     ANNOUNCE PROGRAM TO OPPERATOR
        PRINT TAB(10);"U.S. ARMY-ALASKA MARS MEMBERSHIP FILE SYSTEM"
        PRINT : PRINT : PRINT : PRINT : PRINT


REM - - ARRAY DEFINITION

        MAX.MEMBERS=100
        DIM CALL$(MAX.MEMBERS+1),CODE$(6)

REM - - CLEAR INDEX ARRAY
        FOR I=1 TO MAX.MEMBERS
           CALL$(I)="0"
        NEXT I

REM - - OPEN MEMBER FILE
        MEMBER$=" B:MEMBERS.LIB"
        FILE MEMBER$(129)

REM - - CREATE INDEX IN CORE
        IF END #1 THEN 11
        FOR I=1 TO MAX.MEMBERS
           READ #1,I;DUM$,DUM$,DUM$,CALL$(I),DUM$,DUM$,DUM$,DUM,DUM$
        NEXT I

11      SIZE=I-1
        PRINT " SIZE OF MEMBER FILE IS CURRENTLY ";SIZE

REM - - DETERMINE FUNCTION TO BE PERFORMED
        CODE$(1)=" ADD"
        CODE$(2)=" CHA"
        CODE$(3)="DEL"
        CODE$(4)=" LIS"
        CODE$(5)=" STOP"

10      FOR J=1 TO 1 STEP 0
           PRINT
           PRINT "FUNCTION (ADD,CHA,DEL,LIS,STOP)";
           INPUT FUNC$
           FOR I=1 TO 5
              IF FUNC$=CODE$(I) THEN   \
                 ON I GOTO 20,30,40,50,60
           NEXT I
        NEXT J
50      STOP

REM

REM - - ADDITIONS TO DATABASE

20      FOR I=1 TO 1 STEP 0
                PRINT
                PRINT "CALLSIGN OF NEW MEMBER (0 TO QUIT)"
                INPUT SIGN$
                IF SIGN$="0" THEN   \
                   PRINT   :\
                PRINT "CURRENT MEMBER FILE SIZE IS ";SIZE    :\
                   PRINT   :\
                   GOTO 10
        REM   CHECK FOR DUPLICATE ENTRY
                GOSUB 80
                IF J<>0 THEN   \
                PRINT SIGN$;" ALREADY ON FILE"    :\
                GOTO 202

        REM   GO GET DATA
                GOSUB 70

        REM   FIND EMPTY SPACE IN FILE
                FOR J=1 TO MAX.MEMBERS
                   IF CALL$(J)="0" THEN 201
                NEXT J
                PRINT "NO MORE ROOM IN DIRECTORY"
                GOTO 202

        REM   WRITE NEW ENTRY TO FILE
        201     CALL$(J)=SIGN$
                GOSUB 75

        REM   IF FILE SIZE EXPANDS, WRITE NEW EOF
                IF J>SIZE THEN :\
                   SIZE = J   :\
                   PRINT #1,(SIZE+1);CHR$(26)
202     NEXT I
REM

REM - - CHANGES TO DATABASE

30      FOR I=1 TO 1 STEP 0
                PRINT
                PRINT " CALL SIGN OF MEMBER(0 TO QUIT)"
                INPUT SIGN$
                IF SIGN$="0" THEN    :\
                   PRINT   :\
                   GOTO 10

        REM     LOCATE CALL IN DIRECTORY, IF NOT FOUND J=0
                GOSUB 80
                IF J=0 THEN   \
                PRINT SIGN$;" NOT ON FILE."   :\
                GOTO 303

        REM     GET CURRENT RECORD FOR MEMBER
                GOSUB 85

        REM     GET NEW ENTRIES FROM OPERATOR
                PRINT "FOR EACH CORRECTION RESPOND TO PROMPT"
                PRINT "WITH - ITEM NAME,NEW ENTRY - (0,0 TO QUIT)"
        301     INPUT ITEM$,ENTRY$
                IF ITEM$="0" THEN 302
                IF ITEM$="NAME1" THEN FIRST$=ENTRY$ : GOTO 301
                IF ITEM$="NAME2" THEN MI$ =ENTRY$ : GOTO 301
                IF ITEM$="NAME3" THEN LAST$ =ENTRY$ : GOTO 301
                IF ITEM$=" CALL" THEN SIGN$ =ENTRY$ : GOTO 301
                IF ITEM$="ADR" THEN ADR$ =ENTRY$ : GOTO 301
                IF ITEM$="CITY" THEN CITY$ =ENTRY$ : GOTO 301
                IF ITEM$="STATE" THEN STATE$ =ENTRY$ : GOTO 301
                IF ITEM$="ZIP" THEN ZIP =VAL(ENTRY$) : GOTO 301
                IF ITEM$="HFONE" THEN HFONE$=ENTRY$  :  GOTO 301
                IF ITEM$="WFONE" THEN WFONE$=ENTRY%  :  GOTO 301
                IF ITEM$=" CLASS" THEN CLASS =ENTRY$ : GOTO 301
                IF ITEM$="DATE" THEN EXPIRE =VAL(ENTRY$) : GOTO 301
                PRINT "INVALID ITEM NAME"
                GOTO 301

        REM     WRITE NEW DATA TO FILE
        302     GOSUB 75

        REM     CONFIRM NEW DISK CONTENTS
                GOSUB 95
                GOSUB 90
```

```
        REM     DONE, GO DO ANOTHER
303     NEXT I

REM

REM - - DELETE A RECORD

40      FOR I=1 TO 1 STEP 0
        PRINT
                PRINT "CALL SIGN OF MEMBER TO DELETE(0 TO QUIT)"
                INPUT SIGN$
                IF SIGN$="0" THEN    \
                   PRINT    :\
                   GOTO 10

        REM     FIND CALL IN DIRECTORY, IF NOT FOUND J=0
                GOSUB 80
                IF J=0 THEN    \
                PRINT SIGN$;" NOT ON FILE."    :\
                GOTO 401

        REM     CLEAR CALL SIGN FROM THE RECORD
                TEMP$=SIGN$    :    SIGN$="0"
                GOSUB 75

        REM     CLEAR CALL FROM DIRECTORY
                CALL$(J)="0"

        REM     NOTIFY OPERATOR OF SUCCESFUL REMOVAL
                PRINT TEMP$;" REMOVED FROM RECORD #";J

401     NEXT I
REM

REM - - LIST DATABASE CONTENTS
REM     EITHER ONE MEMBER OR ALL MEMBERS CAN BE LISTED

50      FOR I=1 TO 1 STEP 0
                PRINT
                PRINT "CALLSIGN OF MEMBER TO PRINT (0 TO STOP)"
                INPUT SIGN$
                IF SIGN$="0" THEN    \
                   PRINT    :\
                   GOTO.10

        REM     'ALL' WILL LIST ENTIRE FILE
                IF SIGN$="ALL" THEN    \
                   PRINT CHR$(12);TAB(15);" "    :\
                   FOR J=1 TO SIZE    :\
                         GORUB 85    :\
                      GOSUB 90    :\
                   NEXT J    :\
                   PRINT CHR$(12);TAB(15);" "    :\
                   GOTO 10

        REM     FIND MEMBER IN DIRECTORY
                GOSUB 80
                IF J=0 THEN    \
                PRINT SIGN$;" NOT ON FILE."    :\
                GOTO 502

        REM     PRINT CONTENTS OF MEMBER RECORD
                GOSUB 85
                GOSUB 90

502     NEXT I


REM

REM - - SUBROUTINES

REM - - COLLECT DATA FROM TERMINAL

70      PRINT " LAST NAME,FIRST,MIDDLE INIT";
        INPUT LAST$,FIRST$,MI$

        PRINT " STREET ADDRESS OR PO BOX";
        INPUT ADR$

        PRINT " CITY,STATE,ZIPCODE";
        INPUT CITY$,STATE$,ZIP

        PRINT "HOME FONE$,WORK FONE$";
        INPUT HFONE$,WFONE$

        PRINT "LICENSE CLASS,EXPIRATYON DATE";
        INPUT CLAS$,EXPIRE

        RETURN

REM - - WRITE DATA TO MEMBER FILE

75      PRINT #1,J;FIRST$,MI$,LAST$,SIGN$,ADR$,CITY$,STATE$,ZIP,
                HFONE$,WFONE$,CLAS$,EXPIRE

        RETURN


REM - - LOCATE MEMBER IN DIRECTORY

80      FOR J=1 TO SIZE
           IF SIGN$=CALL$(J) THEN 81
@       NEXT J

        J=0 REM 'NOT FOUND' FLAG

81      RETURN

REM - - GO READ MEMBERS RECORD

85      READ #1,J;FIRST$,MI$,LAST$,SIGN$,ADR$,CITY%,STATE$,
                ZIP,HFONE$,WFONE$,CLAS$,EXPIRE
        RETURN

REM - - PRINT MEMBER DATA

90      PRINT
        PRINT LAST$;",";FIRST$;",";MI$;" - - - ";SIGN$
        PRINT "CLASS - ";CLAS$,"EXPIRES - ";FN.DATE$(EXPIRE)
        PRINT "PHONE NMBRS :  HOME  ";FN.FONE$(HFONE$)
        PRINT TAB(16);"WORK  ";FN.FONE$(WFONE$)
        PRINT " ADDRESS : ";ADR$
        PRINT TAB(11);CITY$;",";STATE$;",";FN.ZIP$(ZIP)
        PRINT


RETIRN
END
```

```
A>RUN MARSBASE
BASIC-E INTERPRETER - VER 2.2

        U.S. ARMY-ALASKA MARS MEMBERSHIP FILE SYSTEM



SIZE OF MEMBER FILE IS CURRENTLY 88

FUNCTION (ADD,CHA,DEL,LIS,STOP)? LIS

CALLSIGN OF MEMBER TO PRINT (0 TO STOP)
? ACMIDLZ

LAPLANTE,F,E - - - ACMIDLZ
CLASS - ADV    EXPIRES - 80/12/02
PHONE NMBRS :  HOME  243-2957
               WORK  274-0263
ADDRESS : 7161 TALL SPRUCE DR
          ANCHORAGE,AK,99502


CALLSIGN OF MEMBER TO PRINT (0 TO STOP)
? 0


FUNCTION (ADD,CHA,DEL,LIS,STOP)? DEL

CALL SIGN OF MEMBER TO DELETE(0 TO QUIT)
? KL7XYZ
KL7XYZ REMOVED FROM RECORD #88

CALL SIGN OF MEMBER TO DELETE(0 TO QUIT)
? 0


FUNCTION (ADD,CHA,DEL,LIS,STOP)? ADD

CALLSIGN OF NEW MEMBER (0 TO QUIT)
? AB7XYZ
LAST NAME,FIRST,MIDDLE INIT? JONES,JOHN,D
STREET ADDRESS OR PO BOX? PO BOSSX 1234-A
CITY, STATE,ZIPCODE? ANCHORAGE,AK,99501
HOME FONE$,WORK FONE$? 1234567,7654321
LICENSE CLASS,EXPIRATION DATE? UNK,811231

CALLSIGN OF NEW MEMBER (0 TO QUIT)
? 0

CURRENT MEMBER FILE SIZE IS 88


FUNCTION (ADD,CHA,DEL,LIS,STOP)? CHA

CALL SIGN OF MEMBER(0 TO QUIT)
? 0


FUNCTION (ADD,CHA,DEL,LIS,STOP)? LIS

CALLSIGN OF MEMBER TO PRINT (0 TO STOP)
? AB7XYZ

JONES,JOHN,D - - - AB7XYZ
CLASS - UNK    EXPIRES - 81/12/31
PHONE NMBRS :  HOME  123-4567
               WORK  765-4321
ADDRESS : PO BOX 1234-A
          ANCHORAGE,AK,99501


CALLSIGN OF MEMBER TO PRINT (0 TO STOP)
? 0


FUNCTION (ADD,CHA,DEL,LIS,STOP)? CHA

CALL SIGN OF MEMBER(0 TO QUIT)
? AB7XYZ
FOR EACH CORRECTION RESPOND TO PROMPT
WITH - ITEM NAME,NEW ENTRY - (0,0 TO QUI1
? HFONE,3217654
? 0,0

JONES, JOHN,D - - - AB7XYZ
CLASS - UNK    EXPIRES - 81/12/31
PHONE NMBRS :  HOME  321-7654
               WORK  765-4321
ADDRESS@: PO BOX 1234-A
          ANCXORAGE,AK,99501


CALL SIGN OF MEMBER(0 TO QUIT)
? 0


FUNCTION (ADD,CHA,DEL,LIS,STOP)? DEL

CALL SIGN OF MEMBER TO DELETE(0 TO QUIT)
? KL7XYZ
KL7XYZ NOT ON FILE.

CALL SIGN OF MEMBER TO DELETE(0 TO QUIT)
? AB7XYZ
AB7XY7 REMOVED FROM RECORD #88

CALL SIGN OF MEMBER TO DELETE(0 TO QUIT)
? 0


FUNCTION (ADD,CHA,DEL,LIS,STOP)? STOP
```

Programmable Calculator Modeling of Experimental Data

By M. Patt, J.R.A. Lemieux, and L. Fisher

INTRODUCTION

At least a part of the excitement and pleasure to be derived from the analysis of random processes stems from the incompleteness and seeming insufficiency of experimental data. A complete set of statistics is seldom, if ever, available to the scientist. Even in the best of laboratory situations, measurements are taken on a discrete (interval) basis and never on a truly continuous basis. In analyzing a process for which a continuous random variable x has been defined, one proceeds experimentally by segmenting the continuous x-axis and observing the percentage time during which the curve is found traversing each segment. If the curve is found in the 3-interval twice the time that it is found in the 2-interval, observation would imply that *three* is twice as likely a value of x as *two*. Continuing in this way, a relative-likelihood curve like the one shown in Figure 2 could be drawn. Although further processing (drawing a smooth curve and normalizing) would yield a probability density curve, such additional processing is not called for here.

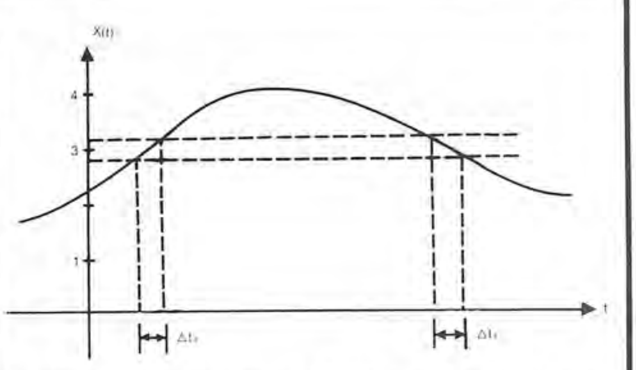


Figure 1. Determination of Statistics Experimentally.

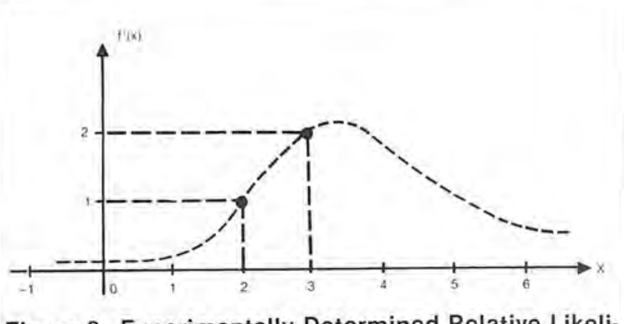


Figure 2. Experimentally Determined Relative Likelihood Curve.

TEXTBOOK MODELS

Statistics which have been obtained experimentally present a number of difficulties from an operational point of view: they cannot be represented nicely by a closed-form expression; they cannot be easily operated upon; and they cannot be readily incorporated into more complex probabilistic expressions. For these and other reasons it is generally desirable to model the experimentally-determined statistics of a random process by a textbook (commonly-used) probability density function.

Density function	First parameter	Second parameter
Normal	μ	σ
Log normal	α	β
Gamma	α	β
Exponential	β	-
Rayleigh	κ	-
Cauchy	α	μ
Beta	a	b

Happily, the selection of probability density function parameters (the α and β of the gamma density function for example) can be facilitated by the use of a programmable pocket calculator. By computing parameter values, textbook density functions can be shaped to match experimentally generated statistics. The fundamental procedure for doing this can be implemented in a few logical steps.

1) Gather experimental data for the random process and organize this data in the form of a relative-likelihood curve. This curve represents the available statistics of the process. The curve need not be normalized.
2) Select a similarly shaped probability density curve (i.e., gamma, exponential, etc.).
3) Load a programmable pocket calculator with the programs for the textbook density function selected. Run the programs in accordance with the keystroke-sequence outlined in this article. Note the density function parameters, as well as the "goodness of fit."
4) Repeat steps 2 and 3 until all similarly shaped density programs have been run.
5) Select the best fitting textbook density function. This five-step procedure and its associated keystroke sequence are applicable without modification to both "reverse-Polish" notation programmable calculators and to "algebraic operating system" notation programmable calculators.

CALCULATOR LAYOUT AND KEYSTROKE SEQUENCE

Common to most current "reverse-Polish" and A.O.S. calculators is the approximate layout of keys depicted in Figure 3.

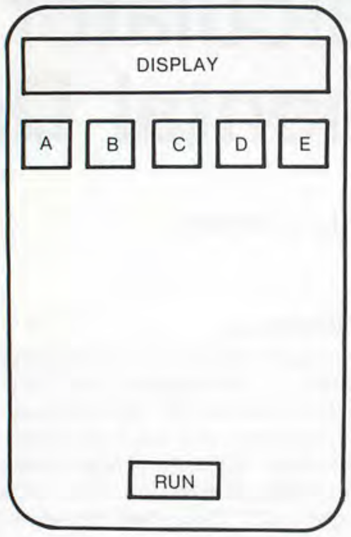


Figure 3. Calculator Keyboard (shown in part).

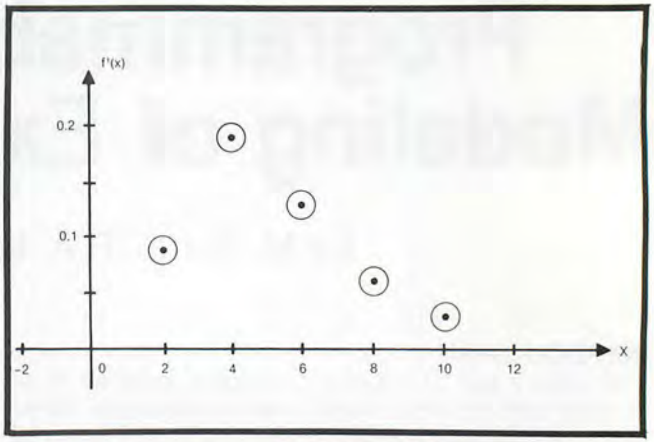
There is only one logical keystroke sequence to be followed. This sequence is embodied in the algorithm described in this section. *It applies to all of the listed textbook density functions, as well as to all programmable calculators.* While the algorithm and keystroke sequence are common to all calculators, the internal programs are not. Programs for both Hewlett-Packard and Texas Instruments calculators are appended to this article.

STEP	DESCRIPTION	KEY	OBSERVE
1	Key in program and reset to first step or load program card	-	-
2	Key in value of x, followed by:	RUN	-
3	Key in value of f(x) followed by:	RUN	-
4	If more data are to be supplied, return to step 2. The minimum data requirement is three (x, f(x)) pairs.	-	-
5	To display the first parameter, press:	A	First Parameter
6	To display the second parameter, if any, press:	B	Second Parameter
7	Enter "Goodness-of-fit" program. This step is omitted for HP67 and SR59 calculators. Press:	RUN	Goodness of Fit
8	To examine the textbook function thus obtained, key in a value of x followed by:	RUN	f(x)
9	Repeat step 8 for as many values of x as are required to obtain a smooth plot of the matching textbook function f(x).	-	-

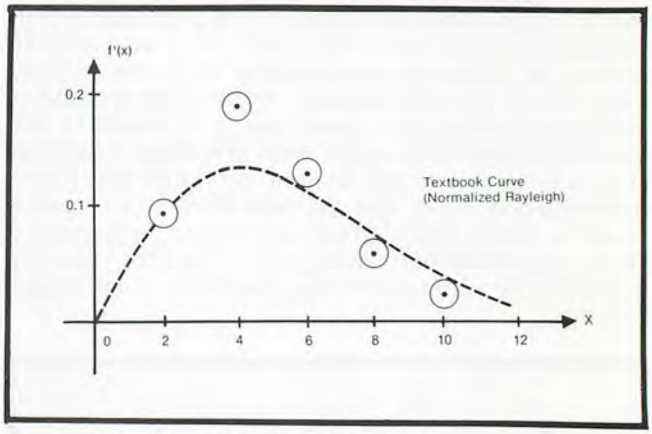
ILLUSTRATIVE EXAMPLE NO. 1

A random process was observed and a relative-likelihood function was determined and sketched. A rayleigh function is to be used to model the density function. Find the parameter "k" of the Rayleigh.

SOLUTION: Entering the Rayleigh program and supplying data points (2,.09), (4,.19), (6,.13), (8,.06), and (10,.03), the



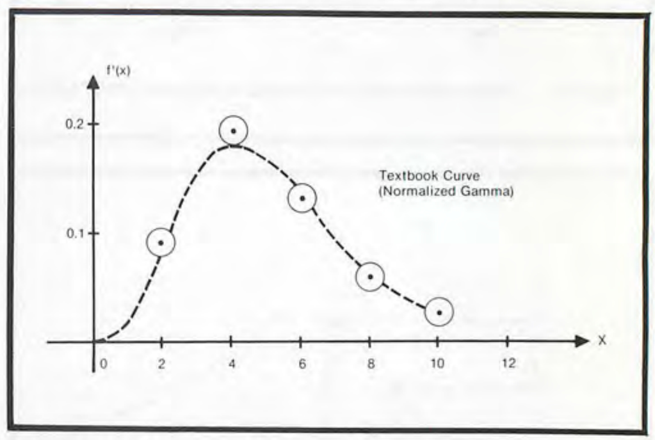
calculator computes and displays the parameter value k = .049. Immediately entering the "Rayleigh Goodness-of-Fit" program, the calculator computes a fit of 0.299 (Best possible fit is 1.000).



ILLUSTRATIVE EXAMPLE NO. 2

A Gamma density function is to be used to model the experimental statistics of Illustrative Example No. 1. Find the parameters α and β of the Gamma.

SOLUTION: Entering the Gamma program and supplying the same data points as in the previous example, the calculator computes parameter values α = 3.488 and β = 1.171 with a goodness of fit of 0.990.

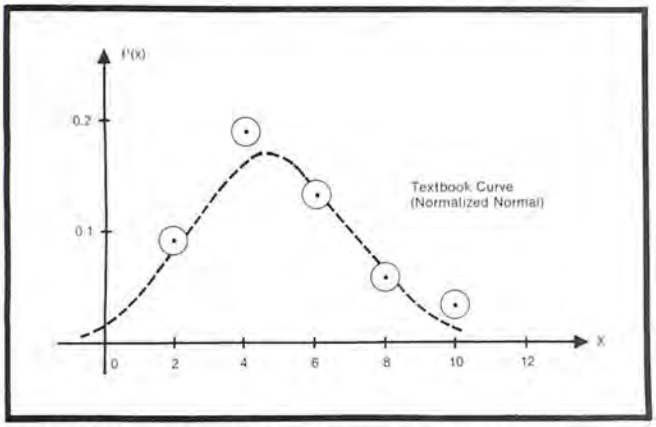


ILLUSTRATIVE EXAMPLE NO. 3

Repeat Illustrative Example No. 1, this time modeling with Normal density function.

SOLUTION: Entering the Normal program and supplying the same three data points as in the previous examples,

the calculator computes parameter values $\mu = 4.857$ and $\sigma = 2.340$ with a goodness of fit of 0.927.



ADVANCED PARAMETER SELECTION

Selection of density parameters has been approached in a direct way — first, the textbook function is selected, and then, the corresponding parameters computed. If one is not committed to a specific textbook function, the procedure may be reversed with surprising results. For a single set of experimental data points, the parameters may be computed and goodness-of-fit obtained for *all textbook density functions.* The goodness-of-fit figures may then be compared with the best-fitting textbook function selected to model the data.

ILLUSTRATIVE EXAMPLE NO. 4

For the experimental data of Illustrative Example No. 1, construct a chart showing several corresponding textbook parameters together with goodness of fit.

SOLUTION:

Textbook Density	Parameters	Fit
Rayleigh	$k = .049$	0.299
Gamma	$\alpha = 3.488, 1.171$	0.990
Normal	$\mu = 4.857, 2.340$	0.927
Cauchy	$\alpha = 3.419, 4.485$	1.00
Log Normal	$\alpha = 1.576, .507$	0.999

Here the Cauchy density function (with parameters $\alpha = 3.419$ and $\mu = 4.485$) would be selected to model the experimentally obtained statistics in the absence of other definitive requirements.

MORE ON NORMALIZATION

To show that the calculator programs are designed to accept both normalized and unnormalized statistical data, a series of two illustrative examples follow. In both cases, the calculator computes textbook density function parameters and generates a normalized density function (with these parameters).

ILLUSTRATIVE EXAMPLE NO. 5

A random process was observed, and the following rates of occurrence found.

x	Rate of Occurrence
2(±.05)	9
4(±.05)	19
6(±.05)	13
8(±.05)	6
10(±.05)	3

A Gamma density function is to be used to model the experimental density function. Find the parameters α and β, as well as the goodness-of-fit.

SOLUTION: Entering the Gamma program and supplying data points (2,9), (4,19), (6,13), (8,6), (10,3), the calculator computes parameter values $\alpha = 3.488$, $\beta = 1.171$. Immediately entering the "Gamma Goodness-of-fit" program, the calculator computes a fit of 0.990. To obtain the normalized Gamma density function with these parameters, one need only supply a value of x and press "RUN" to obtain the corresponding value of $f_X(x)$.

ILLUSTRATIVE EXAMPLE NO. 6

Repeat Illustrative Example No. 5 with the following rates of occurrence.

x	Rate of Occurrence
2(±.05)	0.9
4(±.05)	1.9
6(±.05)	1.3
8(±.05)	0.6
10(±.05)	0.3

SOLUTION: Once again the parameters $\alpha = 3.488$, $\beta = 1.171$ are obtained with a goodness-of-fit of 0.990. The normalized density function obtained by supply x and pressing "RUN" is the same as that obtained in Illustrative Examples 1 and 5.

SUMMARY OF PARAMETER-SELECTION PROCESS

EXPERIMENTAL RELATIVE LIKELIHOOD FUNCTIONS: Determine relative likelihood of occurrence of a few points using the interval technique. Compute associated parameters for all textbook density functions, as well as goodness-of-fit. Select best fitting textbook function and ask the calculator for a normalized plot, if desired.

GENERAL PROGRAM TO MODEL EXPERIMENTALLY OBSERVED RANDOM BEHAVIOR

USING AN HP-67 PROGRAMMABLE POCKET CALCULATOR

(Program begins in location 223)

```
Lbl a
 [R/S]
 ClReg
 P⇄S
 STO C
 [R/S]
 STO D
 [R/S]

Lbl 1
 STO A
 [R/S]
 STO B
Insert #1
 Σ+
 [R/S]
 GTO 1

Lbl A
 P⇄S
 RCL 8
 RCL 4
 RCL 6
 X
 RCL 9
 ÷
 -
 STO 1
 RCL 5
 RCL 4
 x^2
 RCL 9
 ÷
 -
 STO 2
 RCL 1
 x⇄y
 ÷
 STO B
 RCL 6
 RCL 4
 RCL B
 X
 -
 RCL 9
 ÷
 STO A
 RCL 7
 RCL 6
 x^2
 RCL 9
 ÷
 -
 STO 3
 P⇄S
Insert #2
 STO 1
 GTO 2

Lbl B
Insert #3
 STO 2

Lbl 2
 [R/S]
 P⇄S
 RCL 1
 x^2
 RCL 2
 P⇄S
 ÷
 P⇄S
 RCL 3
 P⇄S
 ÷
 [R/S]

Lbl 0
Insert #4
 [R/S]
 GTO 0
```

Normal Density Function

```
Insert #1
 RCL A
 RCL C
 +
 2
 ÷
 RCL D
 RCL B
 ÷
 ln
 RCL A
 RCL C
 -
 ÷

Insert #2
 RCL A

Insert #3
 RCL B
 √x

Insert #4
 RCL 1
 -
 x^2
 CHS
 2
 ÷
 RCL 2
 x^2
 ÷
 e^x
 RCL 2
 ÷
 π
 2
 X
 √x
 ÷
```

Log-Normal Density Function

```
Insert #1
 RCL A
 RCL C
 X
 ln
 2
 ÷
 RCL A
 RCL B
 X
 RCL C
 RCL D
 X
 ÷
 ln
 RCL C
 RCL A
 ÷
 ln
 ÷

Insert #2
 RCL A

Insert #3
 RCL B
 √x

Insert #4
 STO A
 ln
 RCL 1
 -
 x^2
 CHS
 2
 ÷
 RCL 2
 x^2
 ÷
 e^x
 RCL A
 ÷
 RCL 2
 ÷
 π
 2
 X
 √x
 ÷
```

Exponential Density Function

```
Insert #1
 RCL B
 RCL D
 ÷
 ln
 RCL C
 RCL A
 -

Insert #2
 P⇄S
 RCL 7
 STO 3
 RCL 5
 STO 2
 RCL 8
 STO 1
 ÷
 P⇄S

Insert #4
 RCL 1
 ÷
 CHS
 e^x
 RCL 1
 ÷

Insert #3
 0
 ÷
```

Cauchy Density Function

```
Insert #1
 RCL B
 1/x
 RCL D
 1/x
 -
 π
 ÷
 STO 1
 RCL A
 x^2
 RCL C
 x^2
 -
 RCL 1
 ÷
 RCL A
 RCL C
 -
 2
 X
 RCL 1
 ÷

Insert #2
 RCL A

Insert #3
 RCL B

Insert #4
 STO A
 RCL 1
 π
 ÷
 RCL A
 RCL 2
 -
 x^2
 RCL 1
 x^2
 +
 ÷
```

Rayleigh Density Function

```
Insert #1
 RCL C
 RCL B
 X
 RCL A
 RCL D
 X
 ÷
 x^2
 ln
 RCL C
 x^2
 RCL A
 x^2
 -

Insert #2
 P⇄S
 RCL 7
 STO 3
 RCL 8
 STO 1
 RCL 5
 STO 2
 ÷
 P⇄S

Insert #3
 0
 ÷

Insert #4
 STO A
 x^2
 RCL 1
 X
 2
 ÷
 CHS
 e^x
 RCL A
 X
 RCL 1
 X
```

Gamma Density Function

```
Insert #1
 RCL B
 RCL D
 ÷
 ln
 RCL A
 RCL C
 ÷
 ln
 ÷
 RCL C
 RCL A
 -
 RCL A
 RCL C
 ÷
 ln
 ÷

Insert #2
 RCL A

Insert #3
 RCL B
 1/x

Insert #4
 STO A
 RCL 2
 ÷
 CHS
 e^x
 RCL A
 RCL 1
 y^x
 X
 RCL 2
 RCL 1
 1
 +
 y^x
 ÷
 STO A
 RCL 1
 1
 +
 GSB 3
 RCL A
 X⇄Y
 ÷
Insert Sub.
#3 from Beta
program here.
```

Beta Density Function

```
Insert #1
 RCL B
 RCL D
 ÷
 ln
 RCL A
 RCL C
 ÷
 ln
 ÷
 1
 RCL A
 -
 1
 RCL C
 -
 ÷
 ln
 RCL A
 RCL C
 ÷
 ln
 ÷

Insert #2
 RCL A
 1
 +

Insert #3
 RCL B
 1
 +

Insert #4
 STO A
 RCL 1
 RCL 2
 +
 GSB 3
 STO 3
 RCL 1
 GSB 3
 STO÷3
 RCL 2
 GSB 3
 STO÷3
 RCL A
 RCL 1
 1
 -
 y^x
 STOx3
 1
 RCL A
 -
 RCL 2
 1
 -
 y^x
 STOx3
 RCL 3

Subrtn #3
Lbl 3
 1
 -
 STO B
 9
 STI
 +
 ENT
 y^x
 Lstx
 x⇄y
 2
 π
 X
 ^
 X
 √x
 X
 3
 0
 ^
 x^2
 X
 1/x
 1
 -
 1
 2
 ÷
 ^
 ÷
 ^
 +
 e^x
 ÷
Lbl 4
 RCL B
 RCI
 +
 ÷
 DSZ
 GTO 4
 Rtn
```

S.D. SALES CO.

AN EMPIRE IND. CO.

P.O. BOX 28810E DALLAS, TEXAS 75228

S.D. SALES NEW EXPANDABLE EPROM BOARD

16K or 32K EPROM **$49.95 w/out EPROM**

Allows you to use either 2708's for 16K of Eprom or 2716's for 32K of Eprom.

KIT FEATURES:

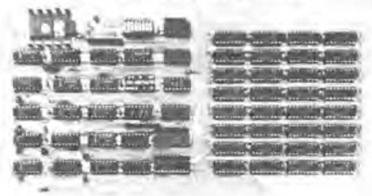
1. All address lines & data lines buffered.
2. Quality plated through P.C. Board, including solder mask and silk screen.
3. Selectable wait states.
4. On board regulation provided.
5. All sockets provided w/board.

WE CAN SUPPLY 450ns 2708's AT $11.95 WHEN PURCHASED WITH BOARD.

32K FOR $475 EXPANDORAM KIT 24K FOR $367.00

MEMORY CAPACITY
MEMORY ADDRESSING
MEMORY WRITE PROTECTION

8K, 16K, 24K, 32K using Mostek MK4115 with 8K boundaries and protection. Utilizes DIP switches. PC board comes with sockets for 32K operation. Orders now being accepted. Allow 6 to 8 weeks for delivery.



INTERFACE CAPABILITY
Control, data and address inputs utilizes low power Schottky devices.

POWER REQUIREMENTS
+8VDC 400MA DC
+18VDC 400MA DC
—18VDC 30MA DC

on board regulation is provided. On board (invisible) refresh is provided with no wait states or cycle stealing required.

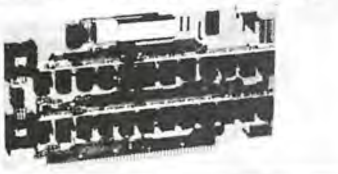
MEMORY ACCESS TIME IS 375ns.
Memory Cycle Time is 500ns.

Buy an S100 compatible 8K Ram Board and upgrade the same board to a maximum of 32K in steps of 8K at your option by merely purchasing more ram chips from S.D. Sales! At a guaranteed price — Look at the features we have built into the board.

16K FOR $259.00 **8K FOR $151.00**

Z-80 CPU BOARD KIT — $139.

CHECK THE ADVANCED FEATURES OF OUR Z—80 CPU BOARD. Expanded set of 158 instructions, 8080A software capability, operation from a single 5VDC power supply, always stops on an M1 state, true sync generated on card (a real plus feature!), dynamic refresh and NMI available, either 2MHZ or 4MHZ operation, quality double sided plated through PC board, parts plus sockets priced for all IC's. *Add $10 extra for Z—80A chip which allows 4MHZ operation. **Z—80 chip with Manual — 29.95**



NEW FROM S.D. "VERSAFLOPPY"™ KIT

THE VERSATILE FLOPPY DISK CONTROLLER

ONLY $149.00

Features: IBM 3740 Soft-Sectored Compatible. S-100 BNS Compatible for Z-80 or 8080. Controls up to 4 Drives (single or double sided). Directly controls the following drives:

1. Shugart SA400/450 Mini Floppy.
2. Shugart SA800/850 Standard Floppy.
3. PERSCI 70 and 277.
4. MFE 700/750.
5. CDC 9404/9406.

34 Pin Connector for Mini Floppy. 50 Pin Connector for Standard Floppy. Operates with modified CP/M operating system and C-Basic Compiler.

The new "Versafloppy" from S.D. Computer Products provides complete control for many of the available Floppy Disk Drives, Both Mini and Full Size. At the heart of "Versafloppy" is the powerful Western Digital FD1771B-1 Single Density Controller Chip. This allows a great flexibility via Control Software. Listings for Control Software are ncluded in the price.

FD 1771B-1 CHIP ALONE $39.95

4K LOW POWER RAM KIT

Fully Buffered — on board regulated — reduced power consumption utilizing low power 21L02 — 1 500ns RAMS — Sockets provided for all IC's. Quality plated through PC board. *Add $10 for 250ns RAM operation.

The Whole Works - $79.95

Low Cost Cassette Interface Kit

Features: Play and record K.C. Standard 2400/1200 Hz tapes, 300 Baud, TTL I/O Compatible, Phase Lock Loop, Both 22 Pin Connector and 8 Pin Molex Connector. Comes partially assembled. Oscillator and phase lock loop pre-tuned to K.C. Standard. Selector switch sends cassette data or auxiliary input data to microprocessor. LED indicates logic 1 level.

$19.95

6 DIGIT ALARM CLOCK KIT

Features: Litronix dual 1/2'' displays, Mostek 50250 super clock chip, single I.C. segment driver, SCR digit drivers. Kit includes all necessary parts (except case). Xfmr optional. Eliminate the hassle.

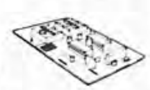
AC XFMR — **$1.50 Case $3.50** **$12.95**

8K LOW POWER RAM — $159.95

Fully assembled and tested. Not a kit. Imsai — Altair — S-100 Buss compatible, uses low power static 21L02-500ns fully buffered on board regulated, quality plated through PC board, including solder mask. 8 pos. dip switches for address select.

DIGITAL LED READOUT THERMOMETER — $29.95

Features: Litronix dual 1/2'' displays. Uses Silicoaix LD131 single chip CMOS A/D converter. Kit includes all necessary parts (except case); AC line cord and power supply included. 0-149° F.



Bowmar 4 Digit LED Readout Array
Full ½'' Litronix Jumbo Dual Digit LED Displays

4 JUMBO .50'' DIGITS ON ONE STICK! WITH COLONS & AM/PM INDICATOR

$3.95

DL 722 - C.C.	DL 728 - C.C.
DL 721 8 C.A.	DL 727 - C.A.
99c	$1.29

MUSICAL HORN

One tune supplied with each kit. Additional tunes — $6.95 each. Special tunes available. Standard tunes now available: — Dixie — Eyes of Texas — On Wisconsin — Yankee Doodle Dandy — Notre Dame — Pink Panther — Aggie War Song — Anchors Away — Never on Sunday — Yellow Rose of Texas — Deep in the Heart of Texas — Boomer Sooner — Bridge over River Kwai.

CAR & BOAT KIT	HOME KIT	Special Design
$34.95	$26.90	Case $3.50

Jumbo LED Car Clock Kit

FEATURES:

A. Bowmar Jumbo .5 inch LED array.
B. MOSTEK — 50250 — Super clock chip.
C. On board precision crystal time base.
D. 12 or 24 hour Real Time format.
E. Perfect for cars, boats, vans, etc.
F. PC board and all parts (less case) inc.

Alarm option — $1.50
AC XFMR — $1.50

$16.95

RAMS

21L02 - 500NS	8/11.50
21L02 - 250NS	8/15.95
2114 — 4K	14.95
1101A — 256	8/$4.00
1103 — 1K	.35
MK 4115 - 8K	15.45
74S 200 - 256	3.95

CPU's

Z—80 includes manual	29.95
Z—80A includes manual	34.95
8080A CPU 8 BIT	11.95
8008 CPU 8 BIT	6.95

PROMS

1702A - 1K - 1.5us	3.95 or 10/35.
2708 - 8K - 450ns	14.95
5204 - 4K	7.95
82S129 — 1K	2.50
2708U 8K signetics 650ns	9.95

COUNTER CHIPS

MK50397 6 Digit elapsed timer	8.95
MK50250 Alarm clock	4.99
MK50380 Alarm chip	2.95
MK50395 6 digit up/dn. count.	12.95
MK5002 4 digit counter	8.95
MK5021-Cal. chip sq. root	2.50

★ ★ ★ ★SUPER FLOPPY SPECIAL★ ★ ★ ★

S. D. SALES,' VERSAFLOPPY S-100 CONTROLLER BOARD PLUS SHUGART SA 400 FLOPPY DISK DRIVE INCLUDING CABLE FOR ONLY

$479.00

MICRO-DIP $1.95

New — Series 2300
The World's Smallest Coded BCD Dual-In-Line Switch! PC Mount
2300 02G BCD 1-2-4-8
2300 12G BCD 1-2-4-8 Compliment

★ ★ JOY STICKS★ ★

FOUR 100 K-OHMS POTS

Ideal for electronic games

$3.95

Z-80 Programming Manual

IN DEPTH DETAIL OF THE Z-80 CPU MICRO-COMPUTER

S. D. SALES SPECIAL $9.95

MICROPROCESSOR CHIPS

8212 - I/O port	3.50
8214 — P.I.C.	12.95
8216 — Non Invert Bus	4.95
8224 — Clock Gen.	4.95
8226 — Invert Bus	3.95
PIO for Z—80	14.95
CTC for Z—80	14.95
8228 Sys. Controller	8.20
8251 Prog. comm. interfact.	10.95
8255 prog. prep. interface	13.50
8820 Dual Line Recr	1.75
8830 Dual Line Dr.	1.75
2513 Char. Gen.	7.50
8838 Quad Bus. Recvr.	2.00
74LS138N — 1/8 decoder	.99
8T97-Hex Tri-State Buffer	1.25
1488/1489 RS232	1.50
TR 1602B Uart	3.95
TR 1863 Uart	8.50
FD 1771B-1	39.95

CMOS

4001	19	4029	99
4002	19	4042	69
4011	19	4047	1.50
4013	32	4049	35
4016	32	4069	23
4017	95	4071	19
4020	97	4076	97
4022	97	14518	1.10
4024	75	14528	85
4027	39	14529	85

* Thermistors 1.5K ohm	5/$1.00
Tantalum Caps 1 mfd. 20VOC P.C. Leads	15/$1.00
Flat Pack IC Assort.	20/$1.00
Electrical Coil 13T Type C - 10T Type C	12/$1.00
2 Transistor Audio	8/$1.00
Trimmer Pots 10K, 20K, 25K, Mini	10/$1.00
Disc Caps For Bypass .01 mfg - 100 WUDC PC Leads	40/$1.00
New Cambion Jacks Part #450-4352 Gold Plated	50/$1.00*
*Silicon Rectifier Special 1N4007. 1 amp 1000 PN.	10/$1.00
Photocell Assortment	12/$1.00
Plastic Readout Filters Amber	6/$1.00
Disc Cap Assortment	60/$1.00
P.C. Lead Diodes 1N4148 1N 914	100/$2.00
1N4002-1A-100 PN	40/$1.00
MICA Trimmer PC402 Miniature 1.5-20 P.C. P.C. Mount	4/$1.00
Resistor Special 22 ohm Carbon Comp.	25/$1.00
*Resistor Assortment 1/4 W 5% & 10% PC leads	200/$1.50*

CHOOSE $1. FREE MERCHANDISE FROM ASTERISK ITEMS ON EACH $15 ORDER

CALL IN YOUR BANKAMERICARD (VISA) OR MASTER CHARGE ORDER IN ON OUR CONTINENTAL TOLL FREE WATTS LINE:

1—800—527—3460

Texas Residents Call Collect:

214/271—0022

DEALER INQUIRIES INVITED!

VISA master charge

NO COD's. TEXAS RESIDENTS ADD 5% SALES TAX. ADD 5% OF ORDER FOR POSTAGE & HANDLING. ORDERS UNDER $10. ADD 75c HANDLING. FOREIGN ORDERS — U. S. FUNDS ONLY!

CIRCLE INQUIRY NO. 92

World Power

By Joe Jaworski

World Power is a war game written in BASIC that is played against the computer. It was written using SWTPC 8K BASIC Version 2.0 but can be easily changed to run on other BASICs, as shown later. The game takes about 20-45 minutes to play, depending on how long you take to plan your strategy; and will run in an 8K buffer area on a SWTPC Computer. The object of the game is to conquer all of the computer's countries by wiping out each country's forces.

The game consists of ten countries:

1) NORTH AMERICA	6) AFRICA
2) AUSTRALIA	7) KOREA
3) SOUTH AMERICA	8) CHINA
4) RUSSIA	9) GREENLAND
5) JAPAN	10) ENGLAND

There are five defenses on each country:

1) TROOPS
2) PLANES
3) ARTILLERY
4) TANKS
5) MISSILES

Each defense may reside on one shore, or all may reside on any shore:

1) NORTH SHORE
2) SOUTH SHORE
3) EAST SHORE
4) WEST SHORE

There are five commands that the program will accept that controls the manipulation and battle status of any country:

1) SHOW COUNTRY STATUS
2) TRANSPORT DEFENSES TO COUNTRIES OR SHORES
3) SHOW GLOBAL STATUS
4) PERFORM A SPY MISSION
5) BEGIN ATTACK

After the program has been loaded and executed, the word "MANEUVER?" should appear. One of the five commands are entered according to the action taken.

SHOW COUNTRY STATUS (1) — The computer will respond with *your* defense status in a particular country. If you choose a country in which you have no defenses, zeroes will appear after each defense is displayed.

SHOW GLOBAL STATUS (3) — The Global Status will display both the quantity of troops that both you (the allies) and the computer (the enemy) have in each country. The last column shows who owns that country (you are started with five). Since the computer shows you the amount of troops, you must then find out what shore they are located on in that country.

PERFORM SPY MISSION (4) — This command will show you the location of all, or part, of the enemy's defenses on any country specified. It costs you ten planes to gain this knowledge, even if no information is gained because the planes got "shot down" (about a 20% probability). For this reason, it is not a good idea to send planes for battle; they are more valuable as a tool for using the spy command. If you have less than ten planes, this command can not be used. The planes can take off from any country and spy on any other country. If you want information about the enemy on the country with which you are in battle, simply type in the same number when the computer asks for the "from" and "to" countries. (See the Sample Run.)

If you don't lose, you'll wind up with a "fight to the finish."

TRANSPORT DEFENSES (2) — This command will transport any amount of each of the five defenses to any shore on any country. To move troops or tanks, for example, on the same country but a different shore, all quantities of that defense must be moved at one time.

BEGIN ATTACK (5) — After all troops and other defenses have been moved into position on other countries, typing in a '5' will start all defenses attacking. The program will continually update the terminal on the present status of each shore under attack. The casualties will be printed out after the battle is over, along with what (if any) defenses were captured. Whenever one side has more than one and a half times of defenses, a battle is over and the winner gains the remaining defenses for its side. For example, if a battle between tanks and artillery broke out, and the remaining defenses were ten tanks for the Allies and sixteen Artillery for the enemy, the enemy will win and your ten tanks would be added to the enemy's defenses. Battles will continue to be displayed as they occur around the world, and will always be printed out until all shores containing both Allied and Enemy defenses are owned by one. Starting attacks on all five countries on all four shores is possible, but it is hard to keep track of so many battles at the same time.

STRATEGY FOR PLAYING

Troops are the most powerful defense there is in the game. It is a good idea when starting the game, to attack the country with the smallest number of troops first. Sending a few spy missions is worth the planes to find out where all the defenses are hiding. If there are many defenses on shore, it is good practice to send many defenses in return. Keep in mind that if you send too many tanks, for example, the excess quantity left over may be captured by the enemy to be used against you during the same attack on the shore. The casualties calculations are based upon the number of defenses residing on the shore under attack. Sending 105,000 troops to destroy 30 tanks; or sending 60,000 troops to destroy 30 tanks will probably cause you to win in *both* cases, but your losses for the larger amount of troops will be much greater than the lesser. Anywhere from none to 33% of your defenses can be lost in the first battle. This is why combining all of your defenses from your five countries and then attacking won't work. You may win the first two or three countries, but your losses will be too great to finish the game. After gaining countries from battle, don't leave your defenses spread all over the world. You may forget about them and lose many battles because you don't have enough defenses to win with a capture.

Since troops are the most powerful of the defenses, avoid going into battle with an even number of troops. If you don't lose, you'll wind up with a "fight to the finish" losing thousands of troops and coming out of the battle not prepared for the next one. It is very difficult to capture troops with any other defense, but defenses may reduce the number of troops in battle from none on up to 1000 troops per attack. Don't reduce the number of troops too greatly by other defenses before attacking; the troops you are killing might be your own.

USING OTHER BASICS

SWTPC's 8K BASIC uses arrays in base 1 (the first element is 1) in all one and two dimensional arrays. Compilers that work in base 0 should have no problems. If problems occur change the appropriate FOR. . .NEXT loops to start at 00. If your compiler/interpreter does not have string concantenation capabilities, replace all occurrences of s$ with the word "SHORE" and remove the plus sign preceding it. To help the user modify or improve the program, the following is a description of the variables and program modules:

VARIABLES

A(10,10) — Allied defenses throughout the world. The X-AXIS element is the country name, and the Y-AXIS elements 1 through 5 are the quantity of defenses. Element 6 through 10 are the location (1-north, 2-south, etc.).

C(10,10) — Same as A(10,10) but the enemy's defenses and locations.
N$(10) — ASCII names of the countries
D$(5) — ASCII names of the defenses
L$(4) — ASCII names of the shores
E — enemy defenses left after each battle
E1 — Allied defenses left after each battle
X,E8,J,D,T,F,Q,L,C,H,K2 — temporary variables

LINES	DESCRIPTIONS
10-130	Initializes all parameters and fills ASCII arrays with names, gets random defenses for Allies and Enemies.
200-230	Main program loop — causes a jump to the appropriate command.
240-282	Prints the status report of the country specified by looking up the data in array A(10,10).
400-570	Transfers defenses to other countries and to other shores.
600-640	Prints a GLOBAL report of all the countries and decides if any enemy troops are residing in countries marked "yes".
1000-1080	Searches through both A(10,10) and C(10,10) to see if both sides are on the same shore in any one country. Returns to main program loop if no shores are occupied by both sides (allies and enemy).
1100-1200 1360-1450	Calculates battle casualties and checks for victory. If Allies have troops in battle, then program goes to 1360. If Enemy has troops, then program branches to 1420.
1270-1348	Reports the victory for this battle.
2000-2080	Calculates the loss of defenses due to transporting.
2500-2550	Calculates the casualties for the present battle.
4000-4010	Prints the end of the game when Allies run out of forces.
5000-5010	Prints the end of the game and the number of maneuvers it took to win.

SAMPLE RUN

```
READY
#RUN
*

MANUEVER? 3


                    --- GLOBAL REPORT ---
COUNTRY        ALLIED TROOPS  ENEMY TROOPS   ALLIED POSSESION?
N. AMERICA      81610          0                   YES
AUSTRALIA       57790          0                   YES
S. AMERICA      55140          0                   YES
RUSSIA          80200          0                   YES
JAPAN           93840          0                   YES
AFRICA          0              58070               NO
KOREA           0              55750               NO
CHINA           0              93670               NO
GREENLAND       0              55400               NO
ENGLAND         0              85570               NO

MANUEVER? 1
COUNTRY? 1


STATUS REPORT: N. AMERICA

DEFENSE        QUANTITY       LOCATION
TROOPS         81610          EAST SHORE
PLANES         61             SOUTH SHORE
ARTILLARY      72             WEST SHORE
TANKS          64             WEST SHORE
MISSILES       45             NORTH SHORE

MANUEVER? 4
ENTER COUNTRIES: (FROM,TO)? 1,6

SPY MISSION REPORT: AFRICA

DEFENSE        QUANTITY       LOCATION
TROOPS          58070          NORTH
PLANES          28             WEST
ARTILLARY       47             EAST
MISSILES        51             SOUTH

MANUEVER? 4
ENTER COUNTRIES: (FROM,TO)? 1,8
```

```
SPY MISSION REPORT: CHINA

DEFENSE          QUANTITY         LOCATION
ARTILLARY         53               EAST
TANKS             51               WEST

MANUEVER? 2
ENTER COUNTRIES: (FROM,TO)
? 1,6
WHAT DEFENSE? 1
QUANTITY? 81610
WHAT SHORE? 1

MANUEVER? 5


BATTLE IN AFRICA ON THE NORTH SHORE!
ENEMY TROOPS     ALLIED TROOPS
58070            81610
57680            65950
45056            58235
33596            52951
31492            44241
21707            34216
17596            27927
14673            27702

***** A VICTORY FOR THE ALLIES IN AFRICA
      ON THE NORTH SHORE

TOTAL CASUALTIES:
ENEMY TROOPS:  43397
ALLIED TROOPS   53908
14673 TROOPS WERE CAPTURED


MANUEVER? 1
COUNTRY? 6


STATUS REPORT: AFRICA

DEFENSE          QUANTITY         LOCATION
TROOPS           42375            NORTH SHORE
PLANES           0
ARTILLARY        0
TANKS            0
MISSILES         0


MANUEVER? 2
ENTER COUNTRIES: (FROM,TO)
? 1,6
WHAT DEFENSE? 2
QUANTITY? 22
WHAT SHORE? 2


MANUEVER? 2
ENTER COUNTRIES: (FROM,TO)
? 1,6
WHAT DEFENSE? 5
QUANTITY? 45
WHAT SHORE? 4

MANUEVER? 5


BATTLE IN AFRICA ON THE SOUTH SHORE!
ENEMY MISSILES  ALLIED PLANES
51              22
37              19

***** A VICTORY FOR THE ENEMY IN AFRICA
      ON THE SOUTH SHORE


TOTAL CASUALTIES:
ENEMY MISSILES:  14
ALLIED PLANES   3


BATTLE IN AFRICA ON THE WEST SHORE!
ENEMY TANKS      ALLIED MISSILES
58               45
39               36
28               31
25               26
20               23
15               21
11               17
8                14

***** A VICTORY FOR THE ALLIES IN AFRICA
      ON THE WEST SHORE
TOTAL CASUALTIES
ENEMY TANKS:  50
ALLIED MISSILES:  31
8 TANKS WERE CAPTURED

MANUEVER? 4
ENTER COUNTRIES: (FROM,TO)? 1,6


SPY MISSION REPORT: AFRICA

DEFENSE          QUANTITY         LOCATION
MISSILES          37               SOUTH

MANUEVER? 4
ENTER COUNTRIES: (FROM,TO)? 1,10
YOU DON'T HAVE ENOUGH PLANES IN N. AMERICA

MANUEVER? 2
ENTER COUNTRIES: (FROM,TO)
? 1,6
WHAT DEFENSE? 3
QUANTITY? 3
WHAT SHORE? 2
```

CIRCLE INQUIRY NO. 60

PROGRAM LISTING

```
0005 REM ** WORLD POWER **
0006 REM ** WRITTEN BY JOE JAWORSKI
0007 REM ** 6/27/77
0008 REM ** WRITTEN FOR SWTPC 8K BASIC REV 2.0
0009 REM **
0010 LINE= 0:W=3:S$=" SHORE"
0020 DIM A(10,10),C(10,10),N$(10),D$(5),L$(5)
0030 DATA N. AMERICA,AUSTRALIA,S. AMERICA,RUSSIA,JAPAN,AFRICA
0040 DATA KOREA,CHINA,GREENLAND,ENGLAND,TROOPS,PLANES,ARTILLARY
0050 DATA TANKS,MISSILES,NORTH,SOUTH,EAST,WEST
0070 FOR X=1TO10:READ N$(X):NEXT X
0080 FOR X=1TO 5:READ D$(X):NEXT X
0090 FOR X=1TO 4:READ L$(X):NEXT X
0100 FOR X=1TO10:C(X,1)=INT((5000*RND)+5000)*10:NEXT X
0110 FOR X=1TO10:FORJ=2TO5:C(X,J)=INT(50*RND)+25:NEXTJ:NEXTX
0120 FOR J=1TO10:FORX=6TO10:LET C(J,X)=INT(4*RND)+1:NEXTX:NEXTJ
0130 FOR X=1TO5:FORJ=1TO10:LET A(X,J)=C(X,J):C(X,J)=0:NEXTJ:NEXTX
0140 PRINT CHR$(27);CHR$(42);TAB(27);"W O R L D  P O W E R":PRINT
0200 REM * MAIN PROGRAM LOOP *
0205  Y6=Y6+1:PRINT:INPUT"MANUEVER",M
0210 IF M>5 THEN200
0220 IF M<1 THEN200
0230 ON M GOTO 240,400,600,800,1000
0240 INPUT "COUNTRY",R
0242 IF R>10 THEN240
0244 IF R<1 THEN240
0250 PRINT :PRINT:PRINT:PRINT"STATUS REPORT: ";N$(R)
0260 PRINT :PRINT"DEFENSE","QUANTITY","LOCATION":FORX=1TO5
0270 PRINT D$(X);TAB(17);A(R,X);TAB(33);
0271 IF A(R,X)<>0 THEN PRINT L$(A(R,(X+5)))+S$:GOTO 282
0280 PRINT
0282 NEXT X:GOTO200
0400 PRINT "ENTER COUNTRIES: (FROM,TO)":INPUT F,T
0410 IF F>10 THEN400
0420 IF F<1 THEN400
0430 IF T>10 THEN400
0440 IF T<1 THEN400
0450 INPUT "WHAT DEFENSE",D
0460 IF D>5 THEN450
0470 IF D<1 THEN450
0480 INPUT "QUANTITY",Q
0490 IF A(F,D)>=Q THEN500
0495 PRINT "YOU ONLY HAVE ";A(F,D);D$(D);" IN ";N$(F):GOTO200
0500 IF F<>T THEN 520
0510 IF A(F,D)<>Q THEN PRINT"YOU MUST MOVE ALL ";D$(D):GOTO200
0520 LET A(T,D)=A(T,D)+Q:A(F,D)=A(F,D)-Q
0530 INPUT "WHAT SHORE",L
0540 IF L>4 THEN530
0550 IF L<1 THEN530
0560 LET A(T,(D+5))=L
0565 IF INT(RND*100)>25 THEN 570
0567 GOSUB 2000
0570 GOTO 200
0600 PRINT :PRINT:PRINT:PRINTTAB(25);"--- GLOBAL REPORT ---"
0610 PRINT "COUNTRY","ALLIED TROOPS","ENEMY TROOPS","ALLIED POSSESION?"
0620 FOR X=1TO10:PRINTN$(X);TAB(18);A(X,1);TAB(34);C(X,1);TAB(55);
0630  K2=0:FORF2=1TO5:IF C(X,F2)<>0 THEN K2=K2+1
0635 NEXT F2
0636 IF K2>0THEN PRINT"NO":GOTO0638
0637 PRINT "YES"
0638 NEXT X
0640 GOTO 200
0800 INPUT "ENTER COUNTRIES: (FROM,TO)",F,T
0810 IF A(F,2)>=10 THEN820
0812 PRINT "YOU DON'T HAVE ENOUGH PLANES IN ";N$(F):GOTO200
0820 LET A(F,2)=A(F,2)-10
0830 PRINT :PRINT:PRINT:PRINT"SPY MISSION REPORT: ";N$(T):PRINT
0840 IF INT(RND*100)>20 THEN 860
0850 IF INT(50*RND)<26 THENPRINT"ALL PLANES LOST AT SEA":GOTO0200
0855 PRINT "PLANES SHOT DOWN OVER ";N$(T):GOTO0200
0860 PRINT "DEFENSE","QUANTITY","LOCATION":FOR X=1TO5
0870 IF INT(RND*100)<40 THEN NEXTX:GOTO 200
0880 PRINT D$(X);TAB(18);C(T,X);TAB(34);
0885 IF C(T,(X+5))<>0 THENPRINTL$(C(T,(X+5))):GOTO0890
0888 PRINT
0890 NEXT X:GOTO 200
1000  K=0:FORX=1TO10:FORJ=1TO5
1010 IF A(X,J)<>0THEN1040
1020 NEXT J:NEXTX:IF K>0THEN1000
1030  H=0:K=0:FORX=1TO10:FORJ=1TO5
1032 IF C(X,J)<>0 THEN K=K+1
1033 IF A(X,J)<>0 THEN H=H+1
1035 NEXT J:NEXTX:IF K=0 THEN 5000
1036 IF H=0 THEN4000
1038 GOTO 200
1040  C=1
1050 IF C(X,C)<>0 THEN1080
1060  C=C+1:IF C=6 THEN 1020
1070 GOTO 1050
1080 IF A(X,(J+5))<>C(X,(C+5)) THEN 1060
1100  K=K+1:PRINT:PRINT:PRINT:PRINTCHR$(7);"BATTLE IN ";N$(X);
1110 PRINT " ON THE ";L$(A(X,(J+5)))+S$;"!"
1120 PRINT "ENEMY ";D$(C),"ALLIED ";D$(J)
1150  O=C(X,C):O1=A(X,J):E=O:E1=O1
1160 IF J=1 THEN IF C<>1 THEN 1360
1162 IF C=1 THEN IF J<>1 THEN 1420
1165 PRINT E,E1
1170 GOSUB 2500
1175 IF E1=0 THEN E8=1:GOTO1270
1178 IF E=0 THEN E8=0:GOTO1270
1180 IF E1>E*1.7 THEN E8=0:GOTO1270
1190 IF E>E1*1.5 THENE8=1:GOTO1270
1200 GOTO 1165
1270 PRINT E,E1:PRINT:PRINT"***** A VICTORY FOR THE ";
1280 IF E8=1 THEN PRINT"ENEMY IN ";:GOTO1288
1285 PRINT "ALLIES IN ";
1288 PRINT N$(X);" ON THE ";L$(A(X,(J+5)))+S$
1300 PRINT "TOTAL CASUALTIES: "
1310 PRINT "ENEMY ";D$(C);":  ";O-E
1320 PRINT "ALLIED ";D$(J);":  ";O1-E1
1330 IF E8=1THENC(X,J)=C(X,J)+A(X,J):A(X,J)=0:GOTO1348
1340  A(X,C)=A(X,C)+C(X,C):C(X,C)=0:A(X,(C+5))=C(X,(C+5))
1345 PRINT E;D$(C);" WERE CAPTURED.":GOTO1020
1348  C(X,(J+5))=A(X,(J+5)):GOTO1020
1360 PRINT E,E1
1370 GOSUB 2500
1375 IF E=0 THEN E8=0:GOTO1270
1377 IF E1=0 THEN E8=1:GOTO1270
1380 IF E1/1000>E THEN E8=0:GOTO1270
1385 IF E>E1*1000 THEN E8=1:GOTO1270
1390 GOTO 1360
1420 PRINT E,E1
1430 GOSUB 2500
1440 IF E=0 THEN E8=0:GOTO1270
1442 IF E1=0 THEN E8=1:GOTO1270
1445 IF E/1000>E1 THEN E8=1:GOTO1270
1448 IF E1>E*1000 THEN E8=1:GOTO1270
1450 GOTO 1420
2000  R=INT(100*RND)+1:X2=INT(10*RND)+2:IFD=1THENX2=INT(100*RND)+2
2010 IF A(T,D)<=X2THEN X2=A(T,D)
2015  A(T,D)=A(T,D)-X2
2020 IF R<33 THEN2070
2030 IF R<66 THEN2080
2040 PRINT "AMBUSH DURING TRANSPORT!"
2050 PRINT "YOU JUST LOST ";X2;D$(D)
2060 PRINT "TOTAL ARRIVED IN ";N$(T);"= ";A(T,D):RETURN
2070 PRINT "SNIPERS ON THE ";L$(A(T,(D+5)))+S$:GOTO2050
2080 PRINT "VIOLENT SEAS SINKS A TRANSPORT SHIP":GOTO2050
2500  E=E-INT((E/3)*RND)-2
2510 IF E<=0 THEN E=0
2520  E1=E1-INT((E1/3)*RND)-2
2530 IF E1<=0 THEN E1=0
2540  A(X,J)=E1:C(X,C)=E
2550 RETURN
4000 PRINT :PRINT:PRINT"SORRY, YOU BLEW IT."
4010 PRINT "ALL YOUR DEFENSES ARE GONE.":PRINT:PRINT:PRINT:GOTO5010
5000 PRINT :PRINT:PRINT:PRINT"CONGRADULATIONS!"
5005 PRINT "YOU HAVE CONQUERED THE WORLD IN ";Y6;"MANUEVERS!"
5010 END
```

CAREER OPPORTUNITIES

Dear Reader:
INTERFACE AGE Magazine, in its attempt to provide you with functional information on a monthly basis, has added this new department to its publication. The purpose of CAREER OPPORTUNITIES is to keep you in touch with those companies that are rapidly expanding and in need of qualified and dependable personnel. It is our goal to assist in advancing the state of the art through communications. Thus, you can look forward to these pages monthly to bring you current career opportunities which are available.

MICRO-MARKET

2708 & 2716
PROM PROGRAMMER
Motorola D1, D2 Kits, MOS Technology KIM-1, or Fairchild F-8 Kit #1
Assembled and Tested $59.95 Each
Optimal Technology, Inc.
Blue Wood 127, Earlysville, VA 22936

tape'n text
PROGRAMMING IN BASIC COURSE.
4 cassette tapes coordinated with 4 printed texts. $19.95. By community college professor. WILLIAMSVILLE PUBLISHING CO., INC. Box 237, Williamsville, NY 14221

SOL or NORTH STAR OWNERS!
VDM-GAMES (require SOLOS or CUTER): Real time ROBOTS and ASTEROID!
PKGV1 (2 Games) – 1200 BAUD CUTS – Tape $20
NORTH STAR BASIC GAMES: ROADRACE, 3D TICTAC, EVENWINS, BIORHYTHM, SUPRWUMP!
PKGN1 (5 Games) on Diskette..$15
MICRO LOGISTICS
P.O. Box 922, Madison Square Station, New York, NY 10010
N.Y. Residents add 8% Sales Tax

NORTH STAR SOFTWARE
Payroll ★ General Ledger
Man. Inventory ★ Retail Inventory
Aged Accounts Receivable
Accounts Payable ★ Mailing List
32K RAM Required
Single or Double Disc
Each System with Many Programs
Price $200 per System or
$1000 for the Set
Write for Documentation
★ SPECIAL ★
Lear Terminal Kit $695
North Star Disckit $599
Micro Computers of New Orleans
1340 S. Carrollton Avenue
New Orleans, Louisiana 70118
(504) 861-8516

INCOME TAX PROGRAM
For Your
TRS-80 SYSTEM!
Computes Federal Return, Form 1040 and Schedules A through F
RUNS ON RADIO SHACK MINIMUM SYSTEM (LEVEL-1 BASIC, 4K RAM, DISPLAY, AND CASSETTE)
CASSETTE & INSTRUCTIONS
$14.95
Advanced Software Division
Contract Services Associates
1846 W. Broadway, Anaheim, CA 92804
(20 Years of Professional Service)

6800 SOFTWARE
INTERRUPT DRIVEN PRINTER ROUTINE — Any BASIC program which uses a printer through a parallel I/O port will run up to 50% faster!........$7.50
GRAPH #2 — Create real-time graphics with ASCII characters on a SWTPC GT-61 display with BASIC! Use standard BASIC statements; no need for tedious machine language programming $5.50
The above 2 programs require SWTPC 8K BASIC 2.0.
LIFE—Play J.H. Conway's game of LIFE. Enter a pattern and watch it evolve! $7.50
POTSHOT—Ballistics trajectory game with a realistic moving display. Every play different! (Requires SWTPC GT-61 display) $7.50
Each program requires a 6800 with MIKBUG and less than 2K of memory. Full source listing and MIKBUG paper tape included, or add $1.00 for Kansas City format cassette.
—Write for flyer on our other programs—
APPLIED MICROCOMPUTER SYSTEMS
Box 68, Silver Lake, NH 03875

DIGITAL GROUP OWNERS
20 MAXI- Game Z Cass. 80 MINI-20
CAT. 1.00
½ REFUND W/O F.A.T.S.Z DEPT 1A
BOX 37 MATTESON, IL 60443

KIM-1 $245
SPECIAL — includes power supply
Immediate Delivery
KIM 4-part harmony music pkg $35
4.5 amp KL Power Supply +5V, +12V $34
Write for list of KIM memory and accessories
A B Computers
P.O. Box 104, Perkasie, PA 18944

Check our prices — We will not be undersold!

POLY-8813	1 drive-No monitor	$2500
POLY-88	System 2	$575
IMSAI I-8080	22 Slot	$575
IMSAI	PC Boards/Kits	15% Off
APPLE II	16K	15% Off
EXTENSYS	Memory	20% Off
NORTH STAR	Disc	15% Off

AAAA Computer How's
1477 Barrington, Suite 17
W. Los Angeles, CA 90025 (213) 477-8478

Affordable Do-it-yourself
MICROCOMPUTER KITS
Build a computer with keyboard, 1K RAM, PC Board, and LED Readout for less than $90. Learn to program it. Information $1. Discounts to dealers.
NBL-F Box 1115 Richardson, TX 75080

SHORTY CASSETTES
3M C-10 Low Noise High Denisty. Short enough to rewind and still hold many programs.
$1.50 ea. — 10 for $14.00 — Send $1.00 for complete supply catalog with refund coupon.
eA ELIJAM ASSOCIATES
24000 BESSEMER
WOODLAND HILLS, CA. 91367

NORTH STAR BASIC PROGRAM
EACH SYSTEM COMPLETE, ON DISKETTE READY TO RUN
* ACCOUNTS RECEIVABLE * GENERAL LEDGER
* ACCOUNTS PAYABLE * PAYROLL
* SOFTWARE LOCATOR * N.S. BASIC TUTORIAL
* BUSINESS STATISTICS * BIORYTHM GENERATOR
* WORD PROCESSING
* FED. INCOME TAX PREP. (AVAIL. 1 JAN. 78)
* RENTAL LOCATOR SERVICE * CHECK BOOK BALANCE
* INSURANCE BILLING * MEDICAL BILLING

--ONLY $25.00 PER SYSTEM, POSTPAID--
EQUIPMENT REQUIRED, SINGLE DRIVE, 8-K FREE MEMORY, PRINTER OPTIONAL
AJA SOFTWARE
AJA SOFTWARE
DEPT. 11 P.O. BOX 2528
ORANGE, CA. 92669

MICRO-MARKET AD RATE:
$50.00 per column inch. Maximum of 4 column inches per ad. Submit ads to:
Micro-Market Ads
INTERFACE AGE Magazine,
PO Box 1234
Cerritos, CA 90701.

COMPUTER HOBBYISTS!
Nationwide Classified Advertising Newsletter
Buy & Sell Hardware & Software
new/used/unique/low-cost
$3.75 for 18 issues.
Free sample issue on request.
ON_LINE Newsletter
24695 Santa Cruz Hwy, Los Gatos, Ca. 95030

FIFO FLEA MARKET

FOR SALE: 3 PT 8KRA at $200 each.; 1 SSM 4K no 2102's at $100; 2 PT 3P + S at $140 each; 1 MITS 88-ACR at $135; 2 DT 2K PROM boards at $45 each; 1 MITS 8800B w/14 connec. A & T $900; 1 ADM-3 A & T $850; 1 Infoton Term. $650; 1 TI Silent 700 Printer, $100 working; 1 NCR PT reader parallel I.F. $200 working. Contact Ken Burgess, 15577 O'Connor, Allen Park, MI 48101. Please include postage.

FOR SALE: ITT 3010 ENVOY Electronic data printer. ASCII coded, RS232 110 baud. Paper tape Reader/Puncher. 8½" paper width, upper and lower case. Full documentation, spare printer and set of spare boards. Weight 120 lbs. Currently used as SWTPC 6800 terminal. $575. Mike Matras, S-4909 Lake Shore Rd., Hamburg, NY 14075, (716) 627-2397.

FOR SALE: 3 ASR33 teletypes with reader control relays for $750.00 each. Edward Winslow, (603) 536-1257.

FOR SALE: 1802 COSMAC Instruction Set Summary, newly arranged in a compact, logical order, typeset on one sheet of paper. Great debugging aid and time-saver. Uses RCA instructions, operations, op-codes, and mnemonics. Available for $1.00 + SASE, from Seth McEvoy, Box 268, E. Lansing, MI 48823.

FOR SALE: North Star Floating Point Board (S-100); 14 digit multiply in 186 microseconds. Factory assembled $250. G. Lyons, 280 Henderson, Jersey City, NJ 07302. (201) 451-2905 eves.

FOR SALE: IBM I/O Selectric Model 72, 2 typeballs, interface parts, $450. INTEL SIM4-01 computer on a board with wire wrap socket - RAM, ROM, 24 I/O lines, manuals, TTY interface, clock reset, CPU $50. SWTPC unfinished CT64 terminal board sets, $200. 24K-16 bit + parity words TI memory boards, 1Kx1 1103 dynamics with on-board sense amps, clocks, etc. 1103's worth $500 + — will sell for $250. Contact Robert Monaghan, P.O. Box 2182 SMU, Dallas, TX 75275.

FOR SALE: Singer 52 line printer 100 LPM, 132 char/line max., connect via I/O channel. Standard paper. Power: 115V, 60Hz; 6 amps. 30"w x 27"d x 38"h. Working and going, shipped "as is" put on line with minimum work. Price inc. manual $650.00. Shipped F.O.B. Cinderella Boot Shop, 104 E. 2nd Ave., Williamson, WV 25661, (304) 235-1995.

FOR SALE: IMSAI 8080 CPU card; or SS music PROM card for 1702's with 2¾K of ROM. Both cards in A-1 shape, fully socketed with full documentation. First $50.00 each takes them. Contact J.B. Ellmers, 841 Kinderkamack Rd., Oradell, NJ 07649, (201) 265-7065 after 6 p.m.

FOR SALE: Processor Technology VDM-1 video display driver w/software, 3P + S interface board, 4KRA static memory board, 2 MITS 4K dynamic memory boards, Ptco mother board w/16 S-100 sockets. All assembled with sockets. All are untested, however. Call or write Bob Stodola, 1910 Beechwood Ave., St. Paul, MN 55116, (319) 698-2731.

FOR SALE: Intel Sim4-02 with sockets and documentation. $125.00. ASR 33 TTY, new covers, working now reproducing tapes. On stand and with plywood shipping crate, $650.00. Altair 680B ready to go, $375.00. Digideck Digital phillips cars. Tape decks and electronics, all documentation. These units look perfect but are untested, $145.00 each. All items plus shipping. Contact Jim Beistle, 3728 Wilkie Way, Ft. Worth, TX 76133.

FOR SALE: Digital Group Phi Deck Digital Tape System (6400 Baud), 20 second access time. Four drives cable case controller board. All factory assembled and tested. Used for three months, working fine when replaced by disks. All documentation and Phi-Mon operating system. Asking $624. Will consider trades, I need a sophisticated terminal. Will ship UPS collect upon confirmation and receipt of $60 deposit. Confirm by phone, (804) 838-1950 Mon-Fri 8 to 5. W.J. McCall, Jr., Tower Box 75, 2101 Executive Dr., Hampton, VA 23666.

FOR SALE: Income Tax Program: On cassette, ready to load into your Radio Shack TRS-80 computer in minimum configuration, i.e., Level I Basic, 4K ROM, Video Display and cassette. Computes 1977 U.S. Form 1040 (individual) Income Tax Return in a few minutes and with absolute accuracy. Figures can be saved on data tape and re-run as desired. Price, each cassette . . . $11.00. Instruction sheet included. Earl J. Bivins, 1846 W. Broadway, Anaheim, CA 92804.

FOR SALE: Assembled and working solid state music PROM board w/o PROMs, $50.00. Assembled and working Southwest Products TB typewriter w/cursot control and UART, $150.00. Contact Webber Hall, Los Angeles, CA, (213) 363-2004 evenings.

FOR SALE: IMSAI 8080 kit with 22 card motherboard, new in box. $545. New MIO board $160. 4K memory board assembled and tested by IMSAI $120. 16K RAM board $350. Contact S. Kim, 10190 Vicksburg, Cupertino, CA 95014, (408) 996-0537 evenings.

SOFTWARE FOR SOL: Palo Alto Tiny Basic source, object, and games on CUTS tape. Write Lichen Wang, 150 Tennyson Ave., Palo Alto, CA 94301.

FOR SALE: DEC VT52 terminal, RS232, 24 linesx80 char., upper and lower case, numeric keypad, 75-9600 baud selectable by switch, direct cursor addressing, limited graphic mode, scrolls automatically upward or downward. Like new. $1400. C. Inglis, 1420-C Post Oak Dr., Clarkston, GA 30021, (404) 938-5549 eves.

PERTEC DISC HELPS DIAGNOSIS

Vectored from Page 75

For routine diagnostic work, the speed and accuracy of diagnosis possible with the Elscint equipment enables as many as 20 patients a day to be tested, depending on the type of work being carried out. The time patients need to spend in the hospital for diagnostic purposes is narrowed down to an average of 2½ hours.

In the past most diagnoses have required exploratory operations, necessitating a 2-day stay in the hospital. Now the majority of patients can attend on an outpatient basis. This creates a better utilization of beds as well as surgical resources.

Patients in countries where medical expenses have to be paid by the patient will benefit from lower hospital treatment bills. Hospitals in countries where medical treatment is paid for by the state will benefit from savings in running costs which could enable the equipment to pay for itself in a matter of months. If, for example, the cost of keeping a patient in the hospital is assumed to be $100 a day, excluding medical fees, and that 75% of the patients for diagnosis are attended on an outpatient basis, then a hospital clinic dealing with fifteen outpatient diagnostic cases a day, which previously would have needed exploratory operations, could theoretically save more than $500,000 a year. The cost of the Elscint equipment complete with Pertec DOS is about $100,000.

FOOTNOTE

Documentation for this article was supplied by Pertec Computer Corporation.

ADVERTISER INDEX

CAST YOUR VOTE

The authors of this issue would like to know your feelings as to which article you consider the best. You may cast your vote on *only* one article. Select the best article and circle the corresponding number on the reader service card.

At the end of the voting period all votes will be tallied, the highest scoring article will be announced and the author will receive a $100 bonus.

ARTICLES

388 Wallet-Size Spaceship Simulator *by James Strickler, John Buffington and Ivy Strickler*

389 LSI-11 Microcomputers in Hospital ICU's *by Ed Heesbeen*

390 Basic Language Cross Assembler *by Peter Reece*

391 Structured Programming in BASIC *by Kamal Shah*

392 Electron Pattern Beam Generator *by Dr. Andrew Wittkower*

393 The WRT System: From Field Notes to Programmable Data *by Dr. Poray-Tucholski*

394 Built a Light Pen *by Scott Fluhrer*

395 Keyboard/Display Interface Peripheral on a Chip *by Larry Grim*

396 Hamming's Error Correction *by Ed Comer*

397 TAPEMON *by Dr. Alan Miller*

398 Random Files Illustrated *by Frederick E. LaPlante, Jr.*

399 Programmable Calculator *by Prof. Martin Patt*

400 World Power — Game *by Joe Jaworski*

OCTOBER WINNER: Assembly Language Structured Programming by Ed Keith
NOVEMBER WINNER: Byte of Music by Christopher G. Smith

READER SERVICE CARD 2

INTERFACE AGE™

February 1978 Issue
Void after May 31, 1978
(Please type or print)

Name ____________________

Title ____________________

Company ____________ Phone (A/C)____________

Address ____________________ ☐ Home ☐ Business

City ____________ State______ Country______ Zip______

ANSWER THE FOLLOWING BY CHECKING *ONE* BOX ONLY PER QUESTION.

1. I Am A
A. ☐ Professional (Medical, Accounting, Law, Etc.)
B. ☐ Engineer (Electronics, Mechanical Etc.)
C. ☐ Business Person (Retail, Wholesale, Etc.)
D. ☐ Educator (Professor, Teacher, Assistant, Etc.)
E. ☐ Student
F. ☐ Hobbyist
G. ☐ Other ________

2. My Status Is
A. ☐ Have All Computing Equipment
B. ☐ Need More Peripherals
C. ☐ Have CPU Only
D. ☐ Have No Equipment
E. ☐ Other

3. I Need This Information For
A. ☐ Immediate Purchase
B. ☐ Purchase 30-60 Days
C. ☐ Comparisons
D. ☐ Literature Library
E. ☐ Other ________

4. My Interest Emphasis Is
A. ☐ Hardware
B. ☐ Firmware
C. ☐ Software
D. ☐ Other ________

5. My Application Is
A. ☐ Hobby Only
B. ☐ Business Only
C. ☐ Hobby & Business
D. ☐ Instruction Purposes
E. ☐ Research
F. ☐ Design & Development
G. ☐ Other ________

6. My *Primary* Source Of "State-of-the-Art" Information Comes From:
A. ☐ Magazines
B. ☐ Exhibits & Conventions
C. ☐ Club Meetings
D. ☐ Direct Mail From Manufacturers.
E. ☐ Other ________

7. I prefer To Buy
A. ☐ Directly From Manufacturers
B. ☐ Local Computer Retailer (Store)
C. ☐ Mail Order
D. ☐ Club Group Purchases
E. ☐ Other ________

8. I Look To *INTERFACE AGE* First For
A. ☐ New Product Information
B. ☐ Software Information
C. ☐ Tutorials
D. ☐ Hardware Articles
E. ☐ Product Advertising
F. ☐ Remarks ________

Please send information on items circled below.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125
126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225
226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250
251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275
276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325
326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350
351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375
376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400

Please use only 1 card per person. Thanks!

READER SERVICE CARD 1

INTERFACE AGE™

February 1978 Issue
Void after May 31, 1978
(Please type or print)

Name ____________________

Title ____________________

Company ____________ Phone (A/C)____________

Address ____________________ ☐ Home ☐ Business

City ____________ State______ Country______ Zip______

ANSWER THE FOLLOWING BY CHECKING *ONE* BOX ONLY PER QUESTION.

1. I Am A
A. ☐ Professional (Medical, Accounting, Law, Etc.)
B. ☐ Engineer (Electronics, Mechanical Etc.)
C. ☐ Business Person (Retail, Wholesale, Etc.)
D. ☐ Educator (Professor, Teacher, Assistant, Etc.)
E. ☐ Student
F. ☐ Hobbyist
G. ☐ Other ________

2. My Status Is
A. ☐ Have All Computing Equipment
B. ☐ Need More Peripherals
C. ☐ Have CPU Only
D. ☐ Have No Equipment
E. ☐ Other

3. I Need This Information For
A. ☐ Immediate Purchase
B. ☐ Purchase 30-60 Days
C. ☐ Comparisons
D. ☐ Literature Library
E. ☐ Other ________

4. My Interest Emphasis Is
A. ☐ Hardware
B. ☐ Firmware
C. ☐ Software
D. ☐ Other ________

5. My Application Is
A. ☐ Hobby Only
B. ☐ Business Only
C. ☐ Hobby & Business
D. ☐ Instruction Purposes
E. ☐ Research
F. ☐ Design & Development
G. ☐ Other ________

6. My *Primary* Source Of "State-of-the-Art" Information Comes From:
A. ☐ Magazines
B. ☐ Exhibits & Conventions
C. ☐ Club Meetings
D. ☐ Direct Mail From Manufacturers.
E. ☐ Other ________

7. I prefer To Buy
A. ☐ Directly From Manufacturers
B. ☐ Local Computer Retailer (Store)
C. ☐ Mail Order
D. ☐ Club Group Purchases
E. ☐ Other ________

8. I Look To *INTERFACE AGE* First For
A. ☐ New Product Information
B. ☐ Software Information
C. ☐ Tutorials
D. ☐ Hardware Articles
E. ☐ Product Advertising
F. ☐ Remarks ________

Please send information on items circled below.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125
126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225
226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250
251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275
276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325
326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350
351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375
376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400

Please use only 1 card per person. Thanks!

READER SERVICE CARD

SUBSCRIPTION

SUBSCRIPTION APPLICATION FORM

PAYMENT MUST ACCOMPANY FORM

SUBSCRIPTION FEE per year: $14.00 U.S., $16.00 Canada/Mexico, $24.00 Foreign.
Must be in U.S. funds drawn on a U.S. bank.

Make check payable to: INTERFACE AGE™ **Magazine**
P.O. Box 1234, Cerritos, CA 90701

Name ____________________ Title ____________

Company ____________________

Home Address ____________________

City ____________ State ________ Country______ Zip ______

Date ____________ Signature ____________________

☐ CHECK ☐ MONEY ORDER ☐ BankAmericard ☐ master charge

Account # ____________________ Expires ____________

PLANNING TO MOVE?

Let us know 8 weeks in advance so that you won't miss a single issue of/INTERFACE.

Attach old label where indicated and print new address in space provided. Also include your mailing label whenever you write concerning your subscription. It helps us serve you promptly.

If you have no label handy, print OLD address here.

Please Print

Name

Company

Address

City State Zip code

Affix Label **Print NEW address here**

Name

Company

Address

City State Zip code

MAIL COMPLETED CARD TO:

INTERFACE™ **Magazine**

P.O. Box 1234
Cerritos, California 90701

MOVING

First Class
Permit No. 217
Clinton, Iowa

BUSINESS REPLY MAIL

No Postage Stamp Necessary if Mailed in the United States

Postage will be paid by

INTERFACE™ **Magazine**

Box 2654

Clinton, Iowa 52732

First Class
Permit No. 217
Clinton, Iowa

BUSINESS REPLY MAIL

No Postage Stamp Necessary if Mailed in the United States

Postage will be paid by

INTERFACE™ **Magazine**

Box 2654

Clinton, Iowa 52732


The Show Company International
presents
Micro
Business '78
March 17, 18, 19, 1978
237
Sponsored by
INTERFACE AGE Magazine
Businessmen, share in the world of advanced computer technology. See how you can save money, increase efficiency and net profit.
MICRO BUSINESS '78, the first totally dedicated small business microcomputer exhibition, will be presenting more than thirty seminars related to using the micro in small business applications. Topics will range from available hardware, to obtaining applications that fit your requirements.
Don't miss MICRO BUSINESS '78. Not just another computer show, but a journey into the world of automation through microcomputers.